BRNO UNIVERSITY OF TECHNOLOGY
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV TELEKOMUNIKACÍ

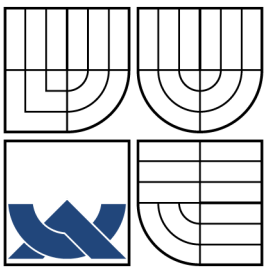
PRIVACY PRESERVING CRYPTOGRAPHIC PROTOCOLS FOR SECURE HETEROGENEOUS NETWORKS

DOCTORAL THESIS
DIZERTAČNI PRÁCE

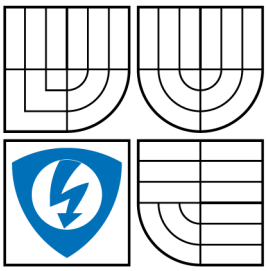
AUTHOR
AUTOR PRÁCE

Ing. LUKÁŠ MALINA

BRNO 2014



BRNO UNIVERSITY OF TECHNOLOGY
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ



FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV TELEKOMUNIKACÍ

PRIVACY PRESERVING CRYPTOGRAPHIC PROTOCOLS FOR SECURE HETEROGENEOUS NETWORKS

KRYPTOGRAFICKÉ PROTOKOLY S OCHRANOU SOUKROMÍ PRO ZABEZPEČENÍ HETEROGENNÍCH SÍTÍ

DOCTORAL THESIS
DIZERTAČNI PRÁCE

AUTHOR
AUTOR PRÁCE
Ing. LUKÁŠ MALINA

SUPERVISOR
VEDOUCÍ PRÁCE
doc. Ing. VÁCLAV ZEMAN, Ph.D.

BRNO 2014

ABSTRACT

The dissertation thesis deals with privacy-preserving cryptographic protocols for secure communication and information systems forming heterogeneous networks. The thesis focuses on the possibilities of using non-conventional cryptographic primitives that provide enhanced security features, such as the protection of user privacy in communication systems. In the dissertation, the performance of cryptographic and mathematic primitives on various devices that participate in the security of heterogeneous networks is evaluated. The main objectives of the thesis focus on the design of advanced privacy-preserving cryptographic protocols. There are three designed protocols which use pairing-based group signatures to ensure user privacy. These proposals ensure the protection of user privacy together with the authentication, integrity and non-repudiation of transmitted messages during communication. The protocols employ the optimization techniques such as batch verification to increase their performance and become more practical in heterogeneous networks.

KEYWORDS

Cryptography, Privacy, Group Signatures, Bilinear Pairing, Optimization, Authentication, Heterogeneous Networks

ABSTRAKT

Disertační práce se zabývá kryptografickými protokoly poskytující ochranu soukromí, které jsou určeny pro zabezpečení komunikačních a informačních systémů tvořících heterogenní sítě. Práce se zaměřuje především na možnosti využití nekonvenčních kryptografických prostředků, které poskytují rozšířené bezpečnostní požadavky, jako je například ochrana soukromí uživatelů komunikačního systému. V práci je stanovena výpočetní náročnost kryptografických a matematických primitiv na různých zařízeních, které se podílí na zabezpečení heterogenní sítě. Hlavní cíle práce se zaměřují na návrh pokročilých kryptografických protokolů poskytujících ochranu soukromí. V práci jsou navrženy celkově tři protokoly, které využívají skupinových podpisů založených na bilineárním párování pro zajištění ochrany soukromí uživatelů. Tyto navržené protokoly zajišťují ochranu soukromí a nepopiratelnost po celou dobu datové komunikace spolu s autentizací a integritou přenášených zpráv. Pro navýšení výkonnosti navržených protokolů je využito optimalizačních technik, např. dávkového ověřování, tak aby protokoly byly praktické i pro heterogenní sítě.

KLÍČOVÁ SLOVA

Kryptografie, ochrana soukromí, skupinové podpisy, bilineární párování, optimalizace, autentizace, heterogenní sítě

MALINA, Lukáš *Privacy preserving cryptographic protocols for secure heterogeneous networks*: doctoral thesis. Brno: Brno University of Technology, Faculty of Electrical Engineering and Communication, Department of Telecommunications, 2014. 166 p. Supervised by doc. Ing. Václav Zeman, Ph.D.

DECLARATION

I declare that I have written my doctoral thesis on the theme of "Privacy preserving cryptographic protocols for secure heterogeneous networks" independently, under the guidance of the doctoral thesis supervisor and using the technical literature and other sources of information which are all quoted in the thesis and detailed in the list of literature at the end of the thesis.

As the author of the doctoral thesis I furthermore declare that, as regards the creation of this doctoral thesis, I have not infringed any copyright. In particular, I have not unlawfully encroached on anyone's personal and/or ownership rights and I am fully aware of the consequences in the case of breaking Regulation § 11 and the following of the Copyright Act No 121/2000 Sb., and of the rights related to intellectual property right and changes in some Acts (Intellectual Property Act) and formulated in later regulations, inclusive of the possible consequences resulting from the provisions of Criminal Act No 40/2009 Sb., Section 2, Head VI, Part 4.

Brno ....

(author's signature)

ACKNOWLEDGEMENT

I would like to thank my supervisor Ing. Václav Zeman, Ph.D. for the guidance and professional support during the work on this thesis. I would like to express gratitude to my family, to my dearest girlfriend, to my colleagues and friends for their patience, understanding and their infinite support.

Brno ....

(author's signature)

Faculty of Electrical Engineering
and Communication
Brno University of Technology
Purkynova 118, CZ-61200 Brno
Czech Republic

http://www.six.feec.vutbr.cz

ACKNOWLEDGEMENT

Výzkum popsaný v této doktorské práci byl realizován v laboratořích podpořených z projektu SIX; registrační číslo CZ.1.05/2.1.00/03.0072, operační program Výzkum a vývoj pro inovace.

Brno . . . . . .

. . . . . .

(author's signature)

EVROPSKÁ UNIE
EVROPSKÝ FOND PRO REGIONÁLNÍ ROZVOJ
INVESTICE DO VAŠÍ BUDOUCNOSTI

CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 INTRODUCTION

Nowadays, there are more and more sophisticated communication and information systems that integrate various types of communication protocols and devices. Emerging technologies such as Wireless Sensor Networks (WSN), Vehicular Ad hoc Networks (VANET), smartgrids, Internet of Things (IoT) or more complex information systems in government infrastructure, large institutions or public transport are widely used and serve to their users. These systems are generally complicated heterogeneous networks with a number of end and intermediate nodes using a wired or wireless connection. Heterogeneous networks connect various types of communication nodes and communication protocols with different specifications. The communication and information systems are usually responsible for data communication, data management and monitoring. On the other hand, these systems can be harmed by a number of passive attacks (e.g. eavesdropping) and active attacks (e.g. data tampering, denial of service attacks, man in the middle attacks, etc.). Hence, the systems must be adequately secured in accordance with possible security threats.

The security and cryptographic protocols used in communication systems are usually designed according to the specific security requirements of the systems. Furthermore, the cryptographic designers have to consider the computational capabilities of intermediate and end nodes, bandwidth, communication delay, the number of users and other aspects as well. Therefore, cryptographic schemes have to meet the properties and requirements of the heterogeneous systems. Besides the security requirements, the cryptographic schemes focus on their computational cost and the length of a cryptographic header used. Nevertheless, there are many services and applications which require not only the basic security properties but also some enhanced properties such as user privacy, user revocation, traceability and so on. Privacy is usually demanded by users especially in services which work with users' private and vital information, e.g. user location, user ID, medical data, etc. On the other hand, there are applications, e.g. vehicular communication, where data confidentiality is not too important like data authenticity, integrity and user privacy during communication among users. In this case, conventional cryptographic schemes and anonymous authentication schemes are not enough, and the designers try to implement a special kind of digital signature schemes that ensure the authentication and integrity of messages and user privacy. These requirements can be provided by group signature schemes. The privacy-preserving solutions usually employ these schemes to ensure user privacy and data security. On the other hand, service providers usually require a mechanism which can detect and correctly identify of a user who violates the rules of the system. This mechanism is usually called as user revocation but some group signature schemes lack this feature.

A comprehensive privacy-preserving and secure communication system requires a set of appropriate cryptographic schemes and primitives that are interconnected to provide not only basic security services but also the advanced properties (e.g. user privacy, user revocation, etc.). Moreover, the privacy-enhancing schemes such as group signature schemes, anonymous authentication schemes, etc., are usually more computationally expensive than conventional cryptographic schemes due to more basic primitives and operations used. Thus, the designers must take into account the different computational specifications of nodes used in heterogeneous networks. This thesis deals with privacy-preserving cryptographic schemes and focuses mainly on research in group signature schemes which are usually based on bilinear pairing operations. In the state of the art, several group signature schemes are proposed by several cryptographers but these schemes are computationally expensive due to many modular arithmetic operations and pairing operations. Due to this fact, these schemes are usually inconvenient for heterogeneous networks with restricted devices. In this thesis, the evaluation of the basic cryptographic primitives and optimization techniques for those schemes are outlined. The thesis mainly contributes by two novel privacy preserving protocols based on group signatures, see Chapters 7 and 8, and by one framework for secure geosocial applications using heterogeneous networks presented in Chapter 9.

The thesis is organized as follows: Chapter 2 presents the field of interest of this thesis. Thesis's objectives and chosen methodology are outlined in Chapter 3. Chapter 4 contains cryptographic background that focuses on the privacy-enhancing technologies, pairing-based cryptography and group signatures. Chapter 5 describes the state of the art in group signatures and other schemes providing the privacy protection in heterogeneous networks. Chapter 6 presents the performance analysis of cryptographic primitives and optimization techniques. Chapters 7 and 8 introduce two novel privacy preserving protocols based on pairing-based group signatures. The first proposed protocol uses group signatures with efficient revocation. The second proposed protocol uses group signatures with categorized batch verification that is designed to be used in the vehicular ad hoc network applications. In these chapters, protocol implementations, evaluations, the comparison of the solutions with related solutions and experimental results are described. Chapter 9 presents the privacy-preserving framework for secure geosocial applications which run on heterogeneous networks. Chapter 10 outlines the discussions dealing with the framework and protocols and the thesis is concluded in Chapter 11.

2 FIELD OF INTEREST AND MOTIVATION

In this chapter, the security issues of heterogeneous networks are discussed. Furthermore, the types of communication networks and devices used in heterogeneous networks are defined. In addition, privacy-enhancing technologies are introduced.

2.1 Security in Heterogeneous Networks

Heterogeneous networks aggregate different types of communication standards, protocols, technologies and end devices. Heterogeneous networks usually cover large areas and contain various wireless networks. In telecommunications research, the heterogeneous network term has two meanings. Firstly, heterogeneous network means the paradigm of seamless and ubiquitous interoperability among various protocols with different types of end nodes. Secondly, heterogeneous network refers to the non-uniform spatial distribution of mobile users in wireless networks.

Nowadays, ubiquitous connectivity services for mobile users are obtained by heterogeneous wireless networks. These networks integrate cellular networks, Mobile Ad hoc Networks (MANET), WLAN, WiMAX, and mesh networks with the non-uniform spatial distribution of users or wireless nodes. In heterogeneous networks, users with their devices can move from one access network to another.

It is essential to consider the security issue as one of the main objectives in developing heterogeneous networks. Currently, standard security approaches provide the basic security properties such as authentication, data integrity, non-repudiation and confidentiality. These security properties can be ensured by common cryptographic primitives and methods which are employed in services provided in heterogeneous networks. There are several security solutions dealing with secure communication, key establishment, authentication or access control in heterogeneous networks, e.g. [174], [85], [172], [30], [75]. On the other hand, user privacy in a heterogeneous environment is a challenge. Due to the open, wireless, distributed and large scale nature, heterogeneous networks are subjects to various attacks. Adversaries have several opportunities to eavesdrop communication, inject data, replay messages, and impersonate others in unsecured heterogeneous networks. Some nodes can be easily used to capture user connections and traffic. This is also a threat to user privacy. Communication applications and services which run on the heterogeneous networks can send sensitive user data like personal identities, activities, location information, movement patterns, financial information, transaction profiles, and so on. The leak of these data can compromise user privacy. Hence, providing user privacy should be one of main objectives in heterogeneous networks.

The security solution of a heterogeneous communication system enhancing user privacy requires basic and several advanced security properties. The main challenge is to find an efficient and secure design of a cryptographic scheme that provides advanced properties and keeps the number of expensive cryptographic operations low. The design of these schemes is more complicated on the computational and memory restricted end nodes which occur in heterogeneous networks. Advanced security properties can be ensured, for example, by group signature schemes. Nevertheless, to ensure the whole spectrum of security properties, the group signature schemes have to be supplemented by other cryptographic tools, such as: anonymous authentication schemes, secure key establishment protocols, ciphers and so on.

This thesis is aimed at end-to-end security at the application layer. The thesis deals with the design of privacy-friendly cryptographic solutions which secures chosen applications and services in heterogeneous networks. The proposed solutions are based on group signature schemes which are convenient for geolocation-based services and services in mobile ad hoc networks, especially vehicular ad hoc network services.

2.1.1 Types of Communication Networks in Heterogeneous Environment

Heterogeneous networks aggregate various communication networks which differ by their scale, the type of link connection (wired, wireless, optical) and communication pattern (peer-to-peer, client-server, master-slave, broadcast, unicast, anycast, multihop, many to one, ad-hoc, infrastructure-based and so on).

Common wired and wireless networks used in heterogeneous environment are characterized in the following text:

- **Ad hoc Network (ANET)** - These wireless networks consist of nodes which can communicate without any base stations. These infrastructure-free networks are self-organized, adaptive and support peer-to-peer multihop communication. Nodes are usually mobile. Mobile Ad Hoc Networks (MANET) allow the spontaneous formation and deformation of mobile networks. Each mobile host acts as a router and the mobility causes route changes. Mobile ad hoc networks face many technical problems such as packet collisions, transmissions errors, routing loops, varied channel quality and node's power consumption and computational restrictions.
- **Mesh Network (MN)** - These wireless networks consist of mobile or stationary mesh clients and mesh routers. A mesh topology is characterized by dynamic self-organization and self-configuration. The connectivity among nodes can be ad hoc. MN uses multiple radios and multiple channels per

radio for increasing the capacity, throughput and lowering interference. The mesh networks usually integrate different wireless technologies, such as IEEE 802.11 (WLAN), IEEE 802.15 (LowPAN), IEEE 802.16 (WiMAX) and so on. A typical Wireless MN (WMN) contains three layers. The first layer connects the access points via wired high-speed connection to Internet. The second layer connects mesh routers via last-mile wireless networks such as Worldwide Interoperability for Microwave Access (WiMAX) networks, which is formed as a multihop backbone. The third layer contains end users or mesh clients that communicate via a multihop connection or via the mesh router in their distance.

- **Cellular Network (CN)** - These wireless networks consist of mobile nodes which communicate via base stations (transceivers). The network is distributed over land areas called cells which use different sets of frequencies in neighboring cells to avoid interference but the same radio frequency can be reused in sufficiently distant areas. There are cells with different sizes and scales such as network—macro, micro, pico, femto, and umbrella cells. Nowadays several cellular technologies are used, for example Global System for Mobile Communications (GSM), General Packet Radio Service (GPRS), Enhanced Data Rates for GSM Evolution (EDGE) and Universal Mobile Telecommunications System (UMTS).
- **Personal Area Network (PAN)** - These networks cover a small area (up to tens meters) and are used for communication among the personal devices such as computers, mobiles, tablets, personal digital assistants, embedded units, smartcards and small hand held devices. These devices can communicate and can be connected via the Internet. Wireless PAN can be realized by wireless network technologies, such as Bluetooth, ZigBee, IrDA, NFC, Z-Wave and others. The special kind of PAN is Body Area Network (BAN) which is based on IEEE 802.15.6 standard. BAN connects body sensor units which serve in healthcare applications, monitoring or in access control services.
- **Local Area Network (LAN)** - These networks cover a limited area such as a home, building, and so on (up to hundreds meters) and are used for communication among the servers and personal devices (computers, laptops, smartphones). LAN provides usually the connection to the Internet. LAN devices may be connected in various topologies such as ring, bus, mesh and star. Local area networks are realized by several technologies and protocols, e.g. Ethernet (802.3) or WiFi (IEEE 802.11).
- **Metropolitan Area Network (MAN)** - These networks cover a metropolitan area (up to thousands meters) and are used for communication among computers, servers and others. MAN (under IEEE 802.6 standard) can be realized

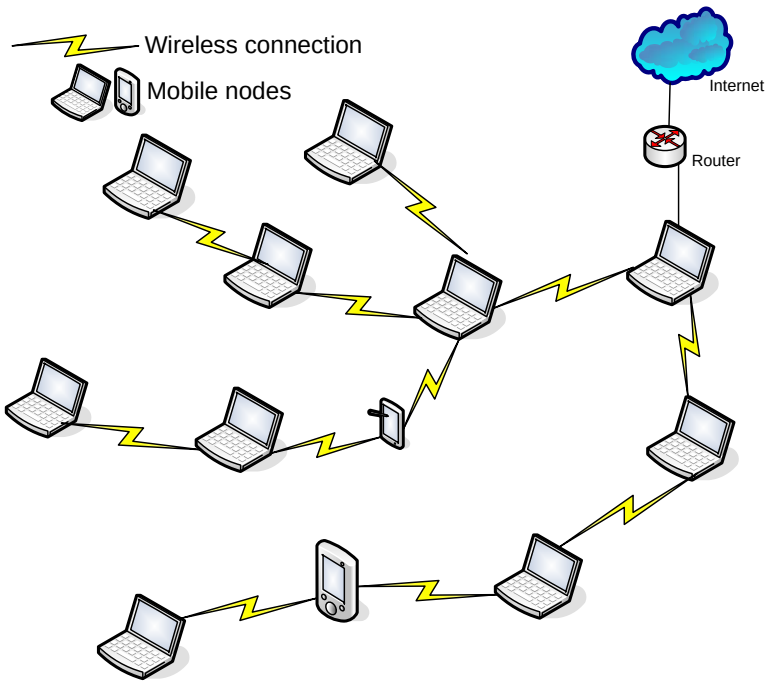


Fig. 2.1: Example Topology of Wireless Mobile Ad hoc Network.

by various technologies such as Distributed-queue dual-bus, Synchronous Optical Network (SONET), Asynchronous Transfer Mode (ATM) and Ethernet with several active and passive network devices.

- **Wide Area Network (WAN)** - These networks cover a broad area (regional or national geographic areas). These networks usually serve as core networks that connect metropolitan and local area networks. The Internet is a kind of WAN. The service providers who manage WAN might employ protocols such as TCP/IP, Synchronous Optical Networking (SONET)/Synchronous Digital Hierarchy (SDH), Multiprotocol Label Switching (MPLS), ATM and Frame relay technology.

2.1.2 Vehicular Ad hoc Networks

Vehicular Ad hoc Networks (VANET) are large scale mobile ad hoc networks (MANETs). VANET use vehicles as mobile nodes to create a mobile communication network. Vehicular ad hoc networks or vehicular networks can be useful in many ways, from increasing a driver safety to reducing traffic congestions. VANET applications can work in short distances, e.g. monitoring collision warnings, change lanes, break alerts and so on. The data processing and communication of these applications should be as fast as possible for the safe and on-time responses of drivers. The sending period of beacon messages should last less than 300 ms [86]. These messages are sent via

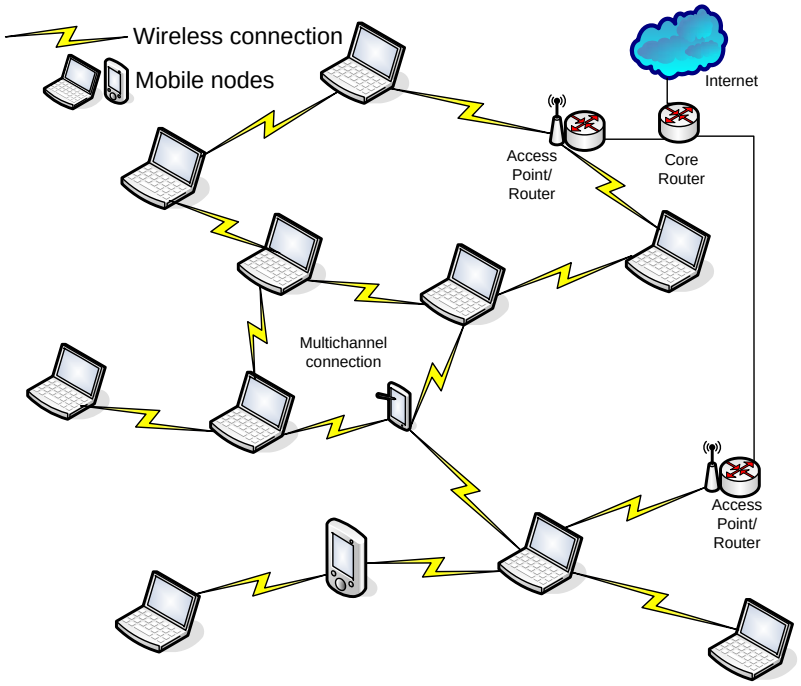


Fig. 2.2: Example Topology of Wireless Mesh Network.

the Vehicle to Vehicle communication (V2V). On the other hand, there are applications which work in wide areas to distribute useful VANET messages, e.g., accident warnings, traffic jam warnings or weather monitoring. These messages can be sent via V2V or via the communication model called Vehicle to Infrastructure (V2I) and then, these messages are broadcasted to other users in a specific area (V2I-I2V). Considering the communication latency due to longer distances, it is assumed that the sending period of these messages should last seconds. In addition, it is also important to provide security and protection against the potential attacks.

2.1.3 Specification of Devices Used in Heterogeneous Networks

Heterogeneous networks consist of communication nodes of different types or architectures. End devices usually have different hardware and software specifications. These devices can provide various computational abilities, memory storages, communication interfaces with specific bandwidth or operation systems with specific software applications. Unfortunately, there are several cryptographic schemes which use the computationally expensive operations such as bilinear pairings or the exponentiation of big numbers. These schemes are not suitable for end devices that are computationally, memory and/or energy restricted. Furthermore, some cryptographic schemes produce a large cryptographic overhead and are not suitable for

narrow bandwidth communication protocols with a small communication overhead. Only some light cryptographic schemes are possible to implement on the restricted devices and restricted connections. These schemes can provide a certain level of security despite of mentioned restrictions.

The devices and their security properties used in the heterogeneous networks are characterized in the following text:

- **RFID/NFC tags** - RFID and NFC tags can serve as authentication items that belong to an entity (a user, a device, an animal, a thing). These tags can be divided to active and passive. Passive tags usually do not support any cryptographic schemes besides some easy-to-run authentication schemes. Active RFID and NFC tags have low computing performance. The CPU frequency reaches units to tens of MHz. On the other hand, the active tags can support several computationally low-cost cryptographic schemes such as symmetric ciphers or message authentication codes [91]. Several works [99], [5] propose to implement privacy enhancing cryptographic solutions on active tags.
- **Smart cards** - Smart cards usually serve as authentication items and/or are used for hosting the services such as e-ticketing in public urban transport, e-payments, e-tolls, access systems or storing the licence key. Smart cards provide low computing performance. The CPU frequency reaches up to several tens of MHz. Smart cards usually offer a sufficient memory storage for various services and applications. Many platforms of smart cards provide conventional cryptographic schemes such as AES, DES, RSA, SHA, MD5, PRNG. Programmable smart cards (.NET, JAVA, Multos) can be used for the implementation of non-conventional cryptographic schemes which can provide advanced security properties. The works [18], [156] describe the implementation of anonymous authentication schemes on JAVA smart cards. Nevertheless, both implementations on the JAVA cards take about 10 s in the verification phase. On the other hand, Multos cards provide modular arithmetic operations natively that speed up the performance of anonymous authentication systems. More results can be found in [129].
- **Smart sensor nodes** - Smart sensor nodes are usually employed in wireless sensor networks. The nodes collect, process and forward data that are sensed in a certain environment. Smart sensor nodes have various performance characteristics. The CPU frequency reaches up to several hundreds of MHz. Nevertheless, the memory capabilities are usually reduced due to minimizing the power consumption. Sensor platforms usually offer some cryptographic schemes, mostly AES and DES. The security solutions in sensor networks have to deal with many restrictions such as key management in mesh topologies,

energy limits, memory limits and so on. There are few works which describe the efficient security solutions, e.g. [166], [135]. Nowadays, it is still an open problem to design the efficient and privacy enhancing security solutions for sensor networks with various nodes and topologies.

- **Hand-held Devices** - Smartphones, tablets, mobiles, etc. can be used as authentication items and/or can host many secure services such as e-payment, geo-localization. Several solutions are described in [3], [162] and [53]. Further, hand-held devices provide various communication interfaces, e.g. GSM, GPRS, UMTS, 802.11/a/b/g/n, NFC etc. Nowadays, these devices have stronger performance characteristics than smart sensors and cards. The CPU frequency reaches up to units of GHz. The operation systems on these devices such as Android, iOS, Windows enable to use various conventional cryptographic methods and schemes. Due to the support of plenty cryptographic and math functions, the hand-held devices can host advanced cryptographic schemes, such as group signatures. On the other hand, the operation systems of these devices are targets for many attacks. Moreover, the secure storage of the secret keys is still a problem these days.
- **Embedded systems control units** - Embedded systems are widely used in industry. But, these units can be also used as on-board units in vehicles and serve as communication/process nodes in Vehicular Ad hoc Networks VANETs. Also embedded systems can be used as smart units in smartgrids networks. The performance characteristic of these devices can be various. In contrast to smart sensors and smartphones, the embedded devices usually do not need a battery source and do not host online services such as smartphones. The CPU frequency reaches several hundreds MHz to units of GHz. The privacy-enhancing security schemes can be implemented on these units. For example, a VANET service needs to broadcast some alerts and notification messages in real time to many nodes (other vehicles, information tables, servers) in a neighborhood. Usually, some VANET services have to deal with tens of messages in real time. The message processing must be as efficient as possible. Further, message security and user privacy have to be solved in VANETs. The works [108] and [179] design security solutions with user privacy for VANETs services. Nevertheless, current solutions are not efficient quite enough if the time interval of message processing is ≤ 300 ms.
- **Computer nodes** - These devices are usually personal computers and servers. The CPU frequency reaches units of GHz, but more cores are usually used. The servers can manage expensive computational procedures and tasks such as generating cryptographic parameters, revocation procedures, etc.
- **Other devices and units** - There are plenty devices which can be used in het-

erogeneous networks but these devices do not participate in the security tasks of cryptographic protocols at the application layer. The security network devices such as firewalls, IPS probes, passive infrared sensors and others devices are out of scope of the proposed cryptographic solutions, and are omitted in this thesis.

2.2 Privacy Enhancing Technologies

Privacy Enhancing Technologies (PETs) are protocols, applications and mechanisms which provide user privacy. PETs prevent unnecessary processing of private information without losing the functionality of information and communication systems. The users of these systems are protected against the leakage of their personally identifiable data and private information. PETs can protect privacy at different layers, mostly at the network layer, transport layer and application layer. General PET methods are described in the next subsections.

2.2.1 Shared Accounts

This simple method can be easily done by creating a bogus online account which is shared by a group of users. An account creator fills in a required information form by bogus data for name, address, phone number, preferences etc. Then, the creator sends the user ID (e.i. Login) and the password to users. Users who use this shared account do not reveal their identities and private information. On the other hand, if an anonymous user misuses the shared account then no party even the creator is able to trace, reveal and revoke this user. This method can be used only in systems where no security is required.

2.2.2 Data Minimization and Blurring

Data minimization methods minimize the private information which are collected in online systems and are used by service providers. These methods try to collect only necessary personal data and ensure the proper function of systems. The collected data are automatically deleted within a certain time period. Users usually negotiate the type of personal data that are sent into the systems. Sometimes, users can inspect, delete and correct their personal data but in compliance with negotiated conditions.

Data blurring is especially used in location-based services. Nowadays, location technologies and services increase in mobile devices, and the actual positions of users can violate their privacy. Blurring the position slightly enhances user privacy and

does not encumber geolocation and geosocial applications. On the other hand, users can not be precisely located due to this approach.

2.2.3 Anonymous Routing and Communication

Online communication becomes interactive and real-time by using ICT technologies such as instant messaging, the world-wide web, remote logins, Voice-Over-IP (VoIP), games, social and geo-social online services. So-called communication anonymizers hide the real online identity such as an email address, an IP address and so on. The real identities are replaced with a non-traceable identities such as one-time email addresses, pseudonyms, a random IP of host from an anonymized network and so on. Anonymization can be done at the network layer and at the application layer. Anonymization at the network layer is usually based on anonymous routing. There are systems such as TOR [58] that provide anonymous routing and protect the privacy of sources. Besides one way anonymous communication, there is bidirectional anonymous communication where the sender and receiver can not be identified. The concrete systems are described in the following text.

Anonymizer Proxies

This simple approach is based on using proxy servers. A proxy server works as a middleman who resends messages from a client to a recipient. The client's source address is replaced by the proxy's source address. Hereby, the proxy server establishes a private channel so the recipient is unable to distinguish who is the initiator of the received message and who sends messages to the proxy server. The proxy server resends the responses back to clients. To increase security, encryption between clients and proxy servers can be added. The proxy server has to be a trusted entity because is able to link sessions and read the content of messages. This property is a weakness of this approach. There is a commercial solution called Anonymizer (https://anonymizer.com/) that provides anonymous web surfing and masks real IP addresses and location.

Crowds

Crowds systems are based on the group of many users. A user can simply get lost in a crowd. Messages are anonymously sent from senders to recipients due to the fact that the messages are relayed by the users of a crowd alliance. The path of the relayed message is random but it also can be direct. An attacker is not able to detect a message originator with a certain probability which is defined by a number of users between a sender and a receiver. A larger group of users provides stronger

anonymity. Users do not have to trust a single entity (e.g. a proxy server). On the other hand, this mechanism burdens communication in the network infrastructure because one message is relayed to several paths. The crowd mechanism can be applied, for example, on web transactions. More information can be found in [146].

Mixes Mechanism

The mixes mechanism called Web Mixes has been introduced in [17]. To improve anonymous communication, this mechanism provides anonymity and unobservability. Unobservability ensures that nobody, not even the transport network, is able to determine who communicates with whom. The Web MIXes system is a structure of several MIX entities, which are controlled by different organizations. A MIX entity can be a simple computer connected via Internet. The MIX entity scrambles and reorders the traffic. Senders encrypt their data of constant size and send it to a MIX entity. The MIX entity receives data from all senders, decrypts and reorders it. Then, these data go via a cascade (chain) of MIX entities. During this process, these MIX entities make some cryptographic operations. The last MIX entity sends the data to a cache-proxy server which communicates with recipients such as web servers. To enhance privacy, the senders can add dummy messages (random data) if they do not have messages to send.

Onion Routing and TOR Protocol

The objective of onion routing is to protect the privacy of the sender and recipient of a message. Messages are relayed by a sequence of network nodes called onion routers that work as proxies. The TOR protocol [58] is a practical implementation that employs onion routing. To prevent eavesdropping, messages are encrypted among the onion routers by using symmetric and asymmetric cryptography. Two nodes, which directly communicate with each other, establish a session key for encryption by asymmetric cryptography (the Diffie-Hellman protocol). To enhance efficiency, the whole correspondence between these nodes is encrypted by symmetric cryptography which uses the session secret keys. With the increase of relay nodes, the layers of encryption increase as well. The message's path is random between the sender and receiver. TOR offers anonymity to communication services using TCP/IP, a low latency, end-to-end integrity and variable exit policies for routers. Nevertheless, onion routing and TOR have weaknesses such as the timing analysis that enable to determine whether a node is communicating with a service due to a low latency. TOR can provide anonymous Internet channels in more complex security solutions used in IP networks.

2.2.4 Anonymous Authentication

Anonymous authentication solves the problem of how to authenticate the users without revealing their identities. Anonymous authentication protocols work on the top of an anonymous routing method, described in previous subsection, that guarantees both user anonymity (removing the identifying information from user sessions) and unlinkability (multiple user's sessions run with a server are indistinguishable). To get more detailed information about the notation please see [109]. Anonymous authentication schemes can be based on the concept of zero knowledge protocols, group signatures, blind signatures, commitments and electronic coins. The electronic coins can be generalized to anonymous tokens or credential systems. A user having a valid token (or credential) can be authenticated and can gain access to protected services.

Anonymous authentication schemes have been proposed in many papers, e.g., [161], [38], [134], [32] [31], [101], [111], [36], [41], [12] [155], [11] and [49]. Anonymous authentication schemes usually use authentication items such as smartcards, protected data storages and so on. These items are used for access to protected spaces (buildings, rooms, labs, etc.). The items can be also plugged to a computer via a reader, and used for access to online services, operation systems and so on. The authentication item which carry authentication data can be combined with user knowledge such as passwords, pin codes, etc.

Attribute-Based Schemes

Attribute-based schemes are cryptographic schemes, e.g. [78], that are designed to enhance user privacy. These schemes provide anonymous proofs of the ownership of personal attributes. The personal attribute represents a specific information about a user, e.g., age, driver license or birthplace. The user who demands a service has to prove the ownership of the attribute to a verifier. The users of an information system that uses an attribute-based scheme are anonymously proven without leaking any other information.

Credential Schemes

Current solutions such as the Idemix scheme [31] and the U-Prove scheme designed by Stefan Brands [27] provide the anonymous authentication of users. The U-Prove scheme uses tokens (a cryptographic construction) to prove user's attributes anonymously. The users' identities are not disclosed when they are presenting attributes. Nevertheless, U-prove does not provide the unlinkability of verification sessions. Therefore, the sessions of the user can be linked together, and this decreases user privacy.

Users of Idemix provide the proofs of credential possessions and the proofs of attribute possessions without revealing their identity. The cryptographic core of Idemix is based on the Camenisch and Lysyanskaya (CL) signature scheme presented in [37]. The scheme used in Idemix is provably secure. Recently, Idemix has been implemented on current smart-card devices (i.e. JavaCards, Multos Cards).

The problem of credential schemes could be the practical revocation of malicious or expired users if slow off-line devices (e.g. smart-cards) are used for storing attributes. Nevertheless, authentication systems based on cards such as the electronic ID cards, employees' smart-cards, library access cards etc. require a practical revocation. The paper [78] presents a novel cryptographic scheme which allows both expiring the user revocation and the de-anonymization of malicious users on commercially available smart-cards. The paper presents results measured on .NET V2+ and MultOS smart-card platforms. On the other hand, the paper does not solve the privacy protection in data communication.

2.2.5 Privacy-Preserving Cryptographic Protocols

These schemes and protocols provide some basic security properties, e.g., confidentiality, data integrity, authentication, non-repudiation and some enhanced security properties that provide user privacy, for example pseudo-anonymity, anonymity, unlinkability, revocation and untraceability. Current cryptographic schemes try to be secure, computationally efficient and keep low communication overhead (short signatures, public keys). But, advanced privacy-preserving properties may increase communication and computational overhead. Besides schemes based on common computational hardness assumptions, such as integer factorization, the RSA problem or the discrete logarithm problem, there are schemes based on elliptic curves and bilinear parings which can offer efficient and new cryptographic schemes.

Privacy-preserving cryptographic protocols usually integrate several types of cryptographic schemes such as authentication schemes, key agreement schemes, digital signatures and encryption schemes. To provide privacy and anonymity to user, the schemes have to be properly combined. Schemes and methods such as group signature schemes, blind signatures, commitment schemes, zero-knowledge proof methods, homomorphic encryption schemes offer several useful privacy-enhancing properties, e.g. identity hiding, binding information, data confidentiality, unlinkability, untracebility, etc.

Currently, several cryptographic schemes providing privacy protection have been designed, such as [134], [32], [12], [11], [111], [101], [31]. These schemes have been applied to authentication and access control solutions with keeping the user privacy. To design security solution in communication systems that ensure user privacy and

data security, a group signature scheme can be a suitable cryptographic tool. Group signature schemes have been introduced by Chaum in [47] and have been studied in, e.g., [39], [15], [38], [20], [26] and [76]. The current drawbacks of group signature schemes are computational expensiveness and long signatures. The basic phases, i.e. signing and verification, of group signature schemes are more expensive than the phases of common digital signature schemes such as RSA, DSA or ECDSA. It is a hard task to implement the group signature schemes onto computationally restricted devices.

This thesis focuses on group signature schemes and their use in privacy-preserving applications in heterogeneous networks. More information about group signature schemes, zero-knowledge protocols and commitment schemes is presented in Chapter 4.

2.2.6 Advanced Requirements on Privacy-Preserving and Secure Communication

Recently, some applications and services require privacy protection inside communication systems. The current secure communication systems offer authentication, data integrity and non-repudiation. But, users and providers of communication systems can demand different security properties which are out of basic security properties. These advanced properties are usually connected with user privacy. The following text sums up the advanced security properties and requirements.

- **Privacy/Anonymity** - privacy protection is ensured for every user in system who follows the rules. Users can communicate in anonymous way. Their identities can be revealed only in special cases, e.g. when a user breaks a rule, authority order, police order, emergency events etc. Privacy protection can be distinguished on two types: a basic anonymity and a full anonymity [26]. the basic anonymity property protects an user identity against passive attacks (e.g. eavesdropping). On the other hand, the full anonymity property protects also against active attacks (e.g. Man in the Middle attack) when an attacker gets access to all old messages, signatures and another data. Nevertheless, if the full anonymity is ensured, then the attacker is not able to connect certain signatures together. This property is called unlinkability.
- **Responsibility/revocation** - every user, who breaks the rules of a system, has to be revealed and revoked. The identification can be done by a certain key (trace key). The revocation ensure that the revoked user has no rights or access in whole systems afterwards. The revocation helps protect the system against repeated misusing. In some applications, the traceability of malicious users' messages is demanded.

- **Efficient and secure key management** - key establishment, key exchange and key revocation in systems have to be secure, efficient and computationally/memory non-expensive. In privacy-preserving solutions, key management has to keep user privacy.
- **Efficient and secure execution of cryptographic protocols** - the phases of a cryptographic protocol should be as efficient as possible to minimize the negative influence of a system, especially, if the restricted devices have been deployed.
- **Exculpability** - no user, either revocation or key manager (group manager, key generator entity, ...), who hold trace keys, can be able to produce a valid signature behalf another user. User can not be accused that makes signature which he does not make. This property is mainly required in group signature schemes. For example, exculpability is ensured in a group signature scheme designed by Boneh, Boyen and Shacham (BBS04) [20].

3 THESIS OBJECTIVES

According to the state of the art, the open problem is to propose a secure, privacy-preserving, and yet efficient solution for communication systems used in heterogeneous networks with varied devices such as sensors, smartphones, embedded system units, smart cards and so on. The several proposals dealing with privacy protection during authentication have been proposed. Nevertheless, there are not many efficient schemes which also ensure privacy protection during communication among huge number nodes. In heterogeneous networks, the end nodes can be devices without any support of cryptographic and math operations. These nodes have to be equipped by software cryptographic libraries that ensure operations needed. Besides software restrictions in heterogeneous networks, the security solutions have to deal also with various bandwidth, type of communication or specific topologies. Key management is another problem related with communication systems which provide a privacy protection. The phases that ensure join of users and their registration have to be secure and maintain privacy protection. Furthermore, it is still problem to design and develop an efficient revocation method which serves in a system with a huge number of revoked users.

3.1 Objectives

The main goal of this dissertation thesis is the research and design of privacy-preserving cryptographic protocols for data communication in heterogeneous networks. Cryptographic protocols must fulfill standard and advanced security requirements, and practically suit the nature of the communication system which may employ devices with low computational power and low memory space, may demand a short time interval for message processing, may have a large number of communication nodes and users, may have restricted bandwidth $\mathbf{B}$ or may have a higher error rate in transmission channels. Whereas efficient anonymous authentication schemes offers only privacy and user authentication, the security solutions in data communication need to keep the data authentication, integrity and non-repudiation of messages that may be sent in short intervals. Group signature schemes should be an useful cryptographic tool which keeps above security properties and ensures the privacy protection. Moreover, the current devices like smartphones and embedded devices with a certain computational performance allow using these more expensive but privacy-preserving schemes. On the other hand, the existed protocols and schemes based on group signatures must be optimized.

This thesis focuses on the design of privacy-preserving cryptographic protocols which use group signatures based on bilinear pairing. The proposed protocols are

designed to ensure chosen standard and advanced security requirements. The main goal of this thesis is to outline the secure and efficient cryptographic protocols based on group signature schemes with an user privacy protection. These protocols are optimized to be proper for applications in heterogeneous networks such as VANET applications and geolocation applications.

The objectives of the dissertation thesis are:

- to analyze and evaluate modern privacy-preserving cryptographic schemes such as group signature schemes and security options in heterogeneous networks with the occurrence of computationally restricted devices,
- to empirically measure the efficiency of cryptographic primitives and operations,
- to propose privacy-preserving cryptographic protocols based on pairing-based group signatures that focus on efficiency and user privacy,
- to optimize the proposed cryptographic protocols in basic phases such as signing the messages and the verification of signatures,
- to propose a privacy preserving cryptographic framework that is suitable for heterogeneous networks and provides user privacy and data security,
- to outline the security analysis of the proposed framework.

3.2 Chosen Methodology

It is important to choose the suitable cryptographic schemes which provide basic and advanced security properties and yet are efficient and applicable for heterogeneous networks. To establish a secure and privacy preserving framework, a key establishment scheme, an encryption scheme and an efficient digital signature scheme with privacy protection have to be properly chosen.

To increase efficiency, the phases of the chosen schemes should be optimized. Thus, it is appropriate to analyze optimization techniques which enable decreasing the number of expensive math and cryptographic operations such as modular exponentiation operations, bilinear pairing operations, etc. Furthermore, the math operations, such as modular multiplication, can be optimized by choosing a proper algorithm. The next step is to design novel cryptographic protocols which ensure the demanded security requirements and are suitable for applications in heterogeneous networks. The analyzed optimization techniques can be applied on the proposed protocols. Due to this step, the certain phases of proposed cryptographic protocols can be more efficient. After this step, the security analyses of proposed cryptographic solutions are outlined. Finally, the chosen phases of the protocols should be implemented, measured, evaluated and compared with related work.

In the following text, the thesis solution procedure is summed up:

1. to analyze and evaluate advanced privacy-preserving methods and cryptographic schemes and choose the schemes providing advanced security properties with a proper cryptographic construction in view of computational and memory requirements,
2. to analyze optimization techniques which are proper to apply on privacy-preserving cryptographic schemes,
3. to propose novel privacy-preserving protocols based on group signatures which ensure the secure and anonymous communication,
4. to optimize the concrete cryptographic phases of pairing-based group signatures and decrease the expensive operations such as modular exponentiation and bilinear pairing by suitable optimization methods,
5. to implement, measure and evaluate the proposed protocols,
6. to propose the complex cryptographic framework providing user privacy protection and security in communication systems formed as heterogeneous networks,
7. to provide the security analyses of the proposed protocols including the active and passive attacks on a system.

4 CRYPTOGRAPHY BACKGROUND AND PRELIMINARIES

This chapter presents the existing cryptographic primitives and schemes that are used in this thesis. Firstly, basic cryptographic primitives, which are used in the proposed security solutions, are introduced. Secondly, pairing-based cryptography, the basic properties of pairing operations and assumptions are described. Finally, group digital signature schemes are introduced, and two pairing-based group signature schemes are outlined. These two schemes are used for proposed protocols in this thesis.

4.1 Basic Cryptographic Primitives

Privacy-preserving cryptographic schemes, e.g. anonymous authentication schemes and group signature schemes, are usually constructed from primitives such as Proof of Knowledge (PK) protocols, Zero-Knowledge (ZK) protocols and commitment schemes.

4.1.1 Proof of Knowledge and Sigma Protocols

Users (provers) can convince a remote party (a verifier) that they know secret numbers (statements) without disclosing them by using a Proof of Knowledge (PK) protocol. This protocol is useful for user authentication and for building the protocols for identity and attribute verification. These types of protocols, e.g. a Proof of Knowledge of Discrete Logarithm (PKDL) protocol, usually have provable properties and their security is based on strong mathematical assumptions. In PKDL, a prover computes the proof of the knowledge of a discrete logarithm. The most common example of a PKDL is the Schnorr's protocol [149], see Figure 4.1. A prover can convince a verifier by using the Schnorr's protocol that they know a secret number w such that $c = g^w \bmod p$ without disclosing w. The environment is the same as for the DSA algorithm, namely p is a large prime, g is the generator of group $\mathbb{Z}_q$, and '$\in_R \mathbb{Z}_q$' denotes a number randomly chosen in the group $\mathbb{Z}_q$ (integers less than q). The usual size of the modulus q is 160 bits and p is at least 1024 bit long. The protocol employs one modular exponentiation and one multiplication.

The Schnorr's protocol is used as the core building block for many modern cryptographic systems, e.g. Proofs of Representation [40], Verifiable Encryption [9] or E-cash schemes [46].

The Schnorr's protocol is a typical Σ (Sigma) protocol because it consists of three steps called commitment, challenge and response. Sigma protocols enable proving

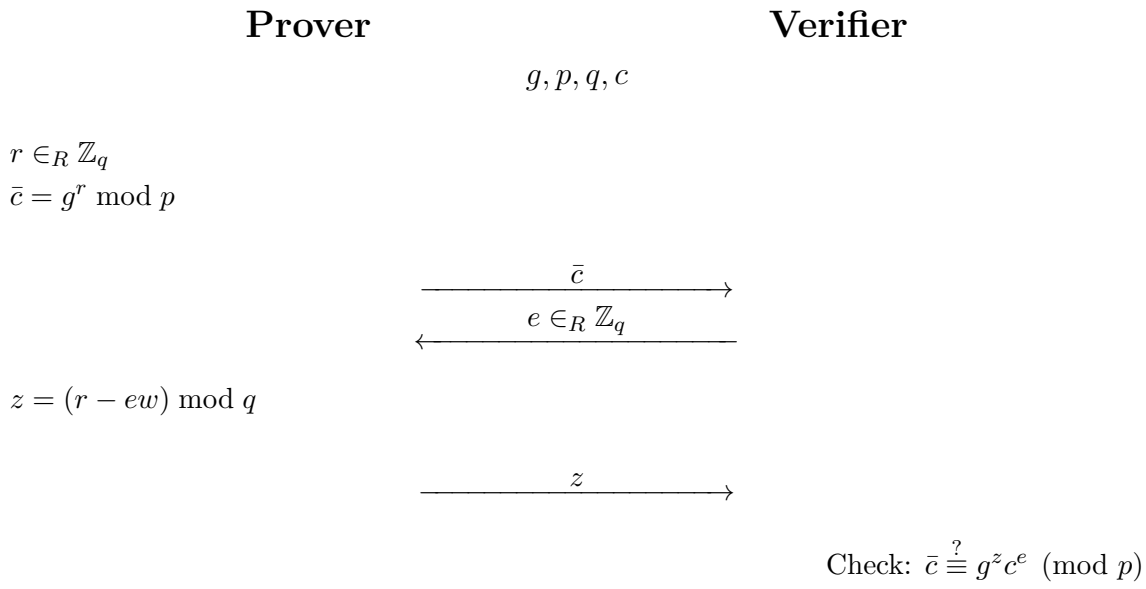


Fig. 4.1: Schnorr's Protocol as Proof of Knowledge of Discrete Logarithm.

various statements such as the knowledge to discrete logarithms. Moreover, the prover can also prove that the discrete logarithm is of a specific form. These protocols are often used in the construction of anonymous and attribute-based schemes.

4.1.2 Zero-Knowledge Protocols

Zero-Knowledge (ZK) protocols or proofs enable that one party (a prover) can convince another party (a verifier) of a given statement is true without yielding any additional information besides the fact that the statement is indeed true. Only the prover who has some secret information is able to prove the statement. After the ZK protocol, the verifier will not be able to prove the statement to anyone else. The formal definitions can be found in [73].

ZK protocols must satisfy these three properties:

- Completeness: in the case of the statement is true, a honest verifier (following the protocol properly) is convinced of this fact by an honest prover.
- Soundness: in the case of the statement is false, a dishonest prover is not able to convince a honest verifier that it is true. The soundness error means that a cheating prover is able to convince the verifier of a false statement.
- Zero Knowledge: in the case of the statement is true, a honest verifier learns nothing else than this fact.

Zero-Knowledge protocols can be divided as follows:

- **Honest Verifier Zero-Knowledge protocols (HVZK)**: protect a prover from a honest verifier. This is a property that the verifier behaves honestly

according to the specified protocol.

- **Computational Zero-Knowledge protocols (CZK)** are protocols that the zero-knowledge condition holds even against cheating verifiers who use a probabilistic polynomial-time Turing machine. The amount of information that the prover leaks to the verifier is negligible.
- **Statistical Zero-Knowledge protocols (SZK)** are protocols that the zero-knowledge condition holds even against cheating verifiers with infinite computing power. The amount of information that the prover leaks to the verifier is negligible.
- **Perfect Zero-Knowledge protocols (PZK)**: protect a prover from a verifier. The protocol is secure even the verifier is dishonest because the amount of information that the prover leaks to the verifier is zero.
- **Interactive Zero-Knowledge protocols (IZK)** are classic ZK protocols where a prover wishes to prove knowledge of secret information to a verifier by communication via a sequence of rounds. In every round, the prover sends the proof to the verifier. For a single round, cheating provers have the 0.5 probability of successfully cheating. Nevertheless, the probability of successfully cheating is negligible by executing a large number of rounds.
- **Non-Interactive Zero-Knowledge protocols (NIZK)**: use a common random string that is shared between a prover and a verifier. The string is sized enough to achieve computational zero-knowledge without requiring an interaction. NIZK protocols enable no interaction between the prover and the verifier. The prover sends the proof to the verifier only once. Pairing-based cryptography can provide powerful and efficient non-interactive zero-knowledge proofs. In pairing-based NIZK protocols, values are usually hidden for the evaluation of the pairing in a commitment. These zero-knowledge proof systems that are using different commitment schemes are built under certain assumptions such as the sub-group hiding and under the decisional linear assumption. The NIZK protocols are also used in proposals in this thesis.

4.1.3 Commitment Schemes

In privacy-preserving cryptography, commitment schemes are two-stage protocols which commit chosen values. These values are hidden to others but the commitment cannot be changed later without changing the hidden values. Stages are called Commit and Reveal. The commitment schemes provide hiding and binding properties and can be interactive or non-interactive. There are perfectly binding and computationally hiding commitment schemes and computationally binding and perfectly hiding commitment schemes. Nevertheless, there is no scheme which is both

perfectly hiding and perfectly binding. The Pedersen commitment scheme [138] is given as an example:

- Setup: A verifier sets large primes p and q such that q divides $p-1$, and the generator g of the order-q subgroup of Z_p^*. Then, a random secret value a is chosen from Z_q and a public value $h = g^a \mathrm{mod} p$ is computed. The values p, q, g, h are public and a is secret.
- Commit: A prover chooses a random value $r \in Z_q$ and commits $x \in Z_q$ by $c = g^x h^r \mathrm{mod} p$. Then, the prover sends c to the verifier.
- Reveal: To open the commitment, the prover reveals x and r, and the verifier verifies that $c = g^x h^r \mathrm{mod} p$

The Pedersen commitment scheme is perfectly (unconditionally) hiding and computationally binding:

- Hiding: for a verifier is statistically indistinguishable to learn x due to the random r which randomizes the commitment c.
- Binding: for a prover is difficult to find (x', r') where $c = g^{x'} h^{r'} \mathrm{mod} p$ is equal $c = g^x h^r \mathrm{mod} p$. The prover has to solve the discrete logarithm problem to compute (x', r'), which is computationally infeasible.

Some zero-knowledge protocols use the commitment schemes to allow the prover to do 'cut and choose' proofs.

4.2 Pairing-Based Cryptography

Pairing-Based Cryptography (PBC) is based on the use of a pairing (mapping) function. Pairings can be a useful and flexible tool to construct new cryptographic protocols, which are often based on new security assumptions.

The main benefit of PBC is that it enables reducing the hardness of one problem in one group to an easier problem in another group (so called gap group). The security of PBC schemes is usually based on another problem which still remains hard. Pairing-based cryptography enables designing three-party protocols. Moreover, PBC schemes use elliptic curves that provide for short lengths of signatures and other cryptographic parameters.

4.2.1 Bilinear pairing operations

A pairing operation is defined by mapping between two elements of certain cryptographic groups G_1 and G_2. The output of the pairing operation is an element of the third cryptographic group. The pairing operation is often described as a bilinear map in the literature. Two types of notation can be usually used: an additive and a multiplicative notation.

Bilinear Pairing with Multiplicative G_1

The multiplicative notation, used in [20], presents the concept of bilinear maps as: G_1, G_2 and G_T are multiplicative cyclic groups of a prime order p. Then, g_1 is the generator of G_1; g_2 is the generator of G_2; and ψ is an isomorphism from G_2 to G_1 where $\psi(g_2) = g_1$. So e is a computable bilinear map $e : G_1 \times G_2 \rightarrow G_T$ with the following properties:

- Bilinearity: for all $u \in G_1, v \in G_2$ and $a, b \in Z_p, e(u^a, v^b) = e(u, v)^{ab}$.
- Non-degeneracy: $e(g_1, g_2) \neq 1_{G_T}$.
- Computability: e is efficiently computable.

Bilinear Pairing with Additive G_1

In the following text, the additive notation is outlined. Let G_1 be a cyclic additive group generated by P, whose order is a prime q, and G_2 be a cyclic multiplicative group with the same order q. Let $e : G_1 \times G_1 \rightarrow G_2$ be a map with the following properties:

1. Bilinearity: $e(aP, bQ) = e(P, Q)^{ab}$ for all $P, Q \in G_1, a, b \in \mathbb{Z}_q$.
2. Non-degeneracy: $e(P, Q) \neq 1$ if $P \neq 0$.
3. Computability: There is an efficient algorithm to compute $e(P, Q)$ for all $P, Q \in G_1$.

A bilinear map satisfying these three properties given above is called a bilinear pairing (more exactly called an admissible bilinear pairing). Typically, the group G_1 is a subgroup of the additive group of points of an elliptic curve E/F_p and the group G_2 is a subgroup of the multiplicative group of a finite field $F^*_{q^2}$. If $G_1 = G_2$ then the pairing is symmetric. If $G_1 \neq G_2$ then the pairing is asymmetric.

Following the notation from [71], the symmetric pairing is a Type 1 pairing. For asymmetric pairings, the function e is called a Type 2 pairing if there is an efficiently-computable isomorphism $\psi : G_2 \rightarrow G_1$. Otherwise, e is called a Type 3 pairing if no efficiently-computable isomorphism from G_2 to G_1 (or from G_1 to G_2) is known.

Pairing operations can be implemented by the Weil pairing, the Tate pairing or other types of pairings (e.g. Ate, Eta, O-Ate,...). Many of them, which compute pairings on pairing-friendly elliptic curves, use the Miller algorithm or the Miller loop [127].

The Weil pairing and the Tate pairing can be used to construct a bilinear pairing between two groups (G_1, G_2) and bilinear maps $e : G_1 \times G_2 \rightarrow G_T$ where G_1, G_2 and G_T are multiplicative groups of prime order p. The Tate pairing is usually faster than the Weil pairing, and hence it is preferred in practice.

Nevertheless, the pairing operation is a costly process and is computationally more expensive than the exponentiation of big numbers used in RSA, curve points respectively (approximately 10 times more, see the paper [151]). More details about the pairing operation efficiency, pairing types and their security can be found in [45].

4.2.2 Hardness Assumptions in Pairing-based Cryptography

In general, the security of cryptographic schemes is based on some hardness assumptions such as the Discrete Logarithm Problem (DLP), the Decision Diffie-Hellman Problem (DDHP) and the Computational Diffie-Hellman Problem (CDHP). These problems can be defined as follows:

- DLP: Given P and aP, it is hard to compute a where P is a point of an elliptic curve over finite field $E(F_q)$, $a \in \mathbb{Z}$.
- DDHP: Given P, aP, bP, and cP, it is hard to decide if $c = ab$.
- CDHP: Given P, aP, and bP, it is hard to compute $(ab)P$.

The Discrete Logarithm Problem in G_1 is no harder than the Discrete Logarithm Problem in G_2. In PBC, two groups are called as gap groups where the DDHP is easy but the CDHP remains hard. Well-known gap groups are those if it is possible to compute pairings: $c \equiv ab \Leftrightarrow e(aP, bP) = e(cP, P)$. In the multiplicative notation, it is easy to determine whether $c \equiv ab$ by checking if $e(g^a, g^b) = e(g, g^c)$.

In PBC, there is another variation of CDHP called the Bilinear Diffie-Hellman Problem (BDHP) with three basic variations:

- general Bilinear Diffie-Hellman Problem (BDHP): For $a, b, c \in \mathbb{Z}_q^*$, given P, aP, bP, cP, compute $e(P, P)^{abc}$.
- Bilinear Inverse Diffie-Hellman Problem (BIDHP): For $a, c \in \mathbb{Z}_q^*$, given P, aP, cP, compute $e(P, P)^{a^{-1}c}$.
- Bilinear Square Diffie-Hellman Problem (BSDHP): For $a, c \in \mathbb{Z}_q^*$, given P, aP, cP, compute $e(P, P)^{a^2c}$.

Some pairing-based schemes are based on extended variations BIDHP and BSDHP. Nevertheless, many of hardness assumptions have been proposed so far, see more in [61], [24]. In the following definitions, only assumptions used in this thesis are listed.

- Strong Diffie-Hellman Problem (q-SDHP): Let G_1, G_2 be two cyclic groups of prime order p, respectively, generated by g_1 and g_2. The q-Strong Diffie-Hellman (q-SDH) problem in G_1, G_2 is defined as follows: Given a (q+3)-tuple of elements $(g_1, g_1^{\gamma}, ..., g_1^{\gamma^q}, g_2, g_2^{\gamma})$ as input, output a pair $g_1^{1\backslash\gamma+x}, x$ where $x \in \mathbb{Z}_q^*$ and the probability is over the random choice of γ and the random bits of `A`.

- External Diffie-Hellman Problem (XDHP): while DDHP is easy in G_2, XDHP states that DDHP is hard in G_1. Let G, generated by g, be a cyclic group of prime order p. The Decisional Diffie-Hellman (DDH) problem in G is defined as follows: Given a tuple of elements (g, g^a, g^b, g^c) as input, output is 1 if $c = ab$ and 0 otherwise.
- Decision Linear Diffie-Hellman problem (DLDHP). Given $u, v, h, u^a, v^b, h^c \in G_1$ as input, the output is 1 if $a + b = c$ and 0 otherwise. More detailed description can be found in [20].

4.2.3 Pairing-Friendly Elliptic Curves

In general, Elliptic Curves Cryptography (ECC) provides the same security level as schemes based on the discrete logarithm problem (e.g. DH) and the factorization problem (e.g. RSA) but with smaller lengths of some parameters (keys, signatures,...). The elliptic curve arithmetic reduces a modular exponentiation operation to a multiplication operation within a group.

In PBC for efficiency reasons, the first pairing argument is set as an elliptic curve $\mathrm{E}(\mathrm{F}_q)[r]$ that is defined over a field F_q, and r is coprime to F_q. Then, the pairing on E is a function e :$\mathrm{E}(\mathrm{K})[r] \times G \rightarrow \mu_r$ where K is a convenient extension of F_q, G is a convenient subgroup of E(K), and μ_r is the subgroup of $\mathrm{F}_q{}^*$ consisting of all r-th roots of unity. Typically, pairing-based cryptography schemes use elliptic curves with a small embedding degree k, e.g., k=2; 3-4; 6-8; 10-16; 12-20, and a large prime-order subgroup with subgroup size r, e.g., r=160 ; 224; 256; 384; 512 bits. The embedding degree of an elliptic curve $\mathrm{E}(\mathrm{K})[r]$ is determined by $r|q^{k-1}$, where K is the smallest extension of F_q containing all coordinates of points of r-torsion of E. PBC schemes use pairing groups based on the Barreto-Naehrig curves [10], e.g., 256-bit with $k = 12$,, supersingular curves with a low embedding degree ($k = 2$), non-supersingular curves called MNT (Miyaji-Nakabayashi-Takano) [128], e.g., 170-bit with $k = 6$, and other pairing-friendly curves which are described in [70]. The supersingular curves and MNT curves offer different performance characteristics.

4.2.4 Pairing-Based Cryptographic Schemes

PBC schemes can be designed in many ways, such as digital signature schemes (short, group, blind, aggregate, ring, threshold), identity-based schemes (for signature, encryption, authentication), a singcryption scheme, etc. For example, the short digital signature scheme BLS [22] is based on bilinear pairings and uses the Weil pairing described in [90]. This scheme produces only 20 B signatures.

More information about pairing-based cryptography can be found in papers [61], [38], [20], [10], [74], [71] or [139].

4.3 Group Signatures

Common digital signatures such as RSA, DSA, ECDSA etc. are widely spread and are used to secure communication and information systems. These schemes can be used in privacy enhancing systems only with pseudonyms for obfuscation of user identity. Nevertheless, the huge number of pseudonymous certifications burdens the key management and user revocation. Moreover, common digital signatures still remain linkable and traceable to a user identity. In order to ensure privacy, authentication and unlinkability of users, a user identity should be decoupled from a verification procedure. Group signatures allow users to authenticate themselves on behalf of a group. Group signatures enable generating pseudonyms by users itself. These pseudonyms are computed by using one secret group member key. All signatures produced by these member keys can be verified by one public group key. Besides the group signatures keep signers (i.e. anonymous group members) in anonymity, another advantage is that there are no more certificates and public keys but only one group public key is used.

Group signatures can be used in many privacy-preserving services and authentication schemes. Group signatures can be understood as a subset of attribute authentication systems which contain only one attribute representing a membership in a group. A user who is a member of a group can sign a message behalf of the group and send the message anonymously to a verifier. A group manager releases only one common group public key gpk which serves to verify the message that is signed by the group member. The basic principle of group signatures is depicted in Fig. 4.2. There are many flavors of group signature schemes with various phases and properties. A general group signature scheme usually contains 6 basic phases: setup (or key generation), join, message signing (or signature generation), signature verification, open and user revocation.

For adversaries, it is computationally hard to solve which member has signed the message. The identities of the members are traceable only in certain circumstances, e.g. breaking the rules. Revocation can be done by the group manager or a revocation manager who owns group manager's secret key $gmsk$. Only the group manager who manages group member secret keys $gsk[i]$ can open the signature of the message and release the real identity of the member. Furthermore, this member can be revoked and his/her signing rights can be abolished. The verifier checks the signature and has to also check if the member has no rights to use services or/and

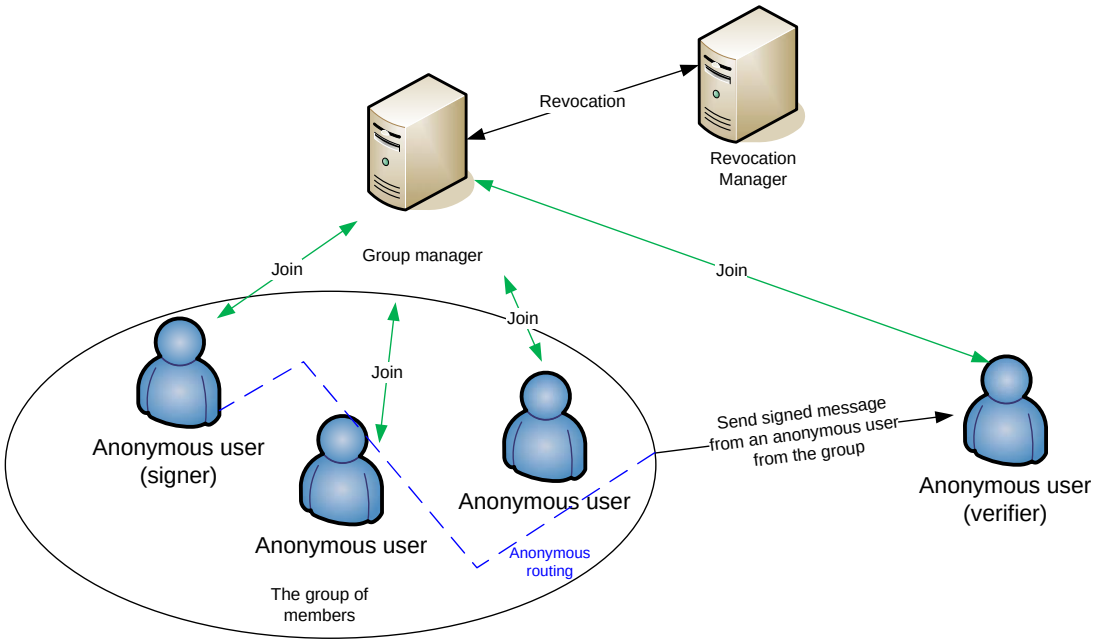


Fig. 4.2: Basic Principle of Group Signatures.

his/her message is of no validity. This phase is called the revocation check. The group signature schemes usually employ the following entities:

- Group manager - this entity adds group members into a group, and generates and issues the secret keys of group members.
- Revocation manager - this entity disclosures the identity of dishonest members.
- User - is a group member who owns the group member secret key $gsk[i]$. The user can sign a message on behalf of the group.
- Verifier - this entity verifies the validity of the signature by using the group public key gpk.

Since the first scheme of a group digital signature was introduced by Chaum and Heyst [47] in 1991, many of group signature schemes have been proposed with various attributes and different ways how to revoke group members. Group signature schemes usually provide the following properties:

- Unforgeability - only an unrevoked group member can create a valid signature on behalf of the group.
- Anonymity - a verifier is not able to determine the identity of a signer.
- Complete anonymity - if an attacker obtains a valid signature and knows gpk and all keys of group members' $gsk[i]$, he is not able to determine the identity of a signer.
- Traceability - all members can be tracked by the group manager or the revocation manager by member's signed message.
- Untraceability - any member cannot be tracked by a verifier and/or other

group members by his/her signed messages.

- Unlinkability - a verifier and other members are not able to link two signatures which have been signed by one member of the group.
- Coalition-resistance - it is impossible to create a valid signature by a subgroup of users.
- Exculpability - even group manager is not able to create the valid signature of a group member.
- Correctness - every correct signature of a group member has to be always accepted during verification.
- Revocation - a revoked member is not able to create valid signatures on behalf of the group.
- Differentiation of group members - all members of a group must have a different $gsk[i]$.
- Immediate-revocation - if a group member is revoked, his capability of creating the group signatures is immediately disabled.

4.3.1 Static and Dynamic Group Signatures

Group signatures can be divided on static and dynamic. In static group signature schemes, the number of group members is fixed during setup (an initialization stage), and group member secret keys are computed for each member in setup. The static group signature schemes have usually 4 phases: key generation, signing, verification and open. These types of group signature schemes do not have a join phase and are not convenient for systems where the number of group members is unpredictable. On the other hand, dynamic group signature schemes have a join phase which enables adding new members into a group. The key generation phase does not produce group member secret keys as in static group signature schemes. The dynamic group signature schemes consist of 5 phases: key generation, join, signing, verification and open. Furthermore, some group signature schemes provide new phases such as a membership revocation and a update procedure. These phases prevent former group members and malicious members, who are excluded from the group, from generating the valid signatures.

4.3.2 Pairing-Based Short Group Signatures

At Crypto 2004, Boneh, Boyen and Shacham proposed a short group signature scheme (BBS04) [20] which is based on the use of bilinear pairing. The security of the BBS04 scheme is based on the Strong Diffie-Hellman and the Decisional Linear assumptions. The scheme uses the standard generalization of Schnorr's protocol

for proving the knowledge of a discrete logarithm. These types of signatures are often called signatures of knowledge due to using a proof of knowledge via the Fiat-Shamir heuristic, see [1], [68]. The scheme is secure in the random oracle model due to employing a hash function $H : \{0,1\}^* \to Z_p$. This group signature scheme consists of these phases: Key generation, Sign, Verify and Open.

Key Generation Phase

In this phase, a generator g_2 in a multiplicative cyclic group G_2 is uniformly selected at random. Then, $g_1 \leftarrow \psi(g_2)$ is set where ψ is a computable isomorphism from G_2 to G_1. Let the bilinear groups G_1, G_2 be subgroups of the group of points of an elliptic curve E/F_q, the trace map on the curve is used as this isomorphism. Parameters $\xi_1, \xi_2 \in Z_p^*, h \in G_1^*$ are randomly selected and $u, v \in G_1^*$ are set such that $u^{\xi_1} = v^{\xi_2} = h$.

Further, a random $\gamma \in Z_p^*$ is selected, and $w = g_2^{\gamma}$ is computed. For each member (the i-th user of the group) a tuple (A_i, x_i) is computed by the holder of γ (a private-key issuer) such that $A_i = g_1^{\frac{1}{x_i+\gamma}}$ where $x_i \leftarrow Z_p^*$ is randomly selected. The tuple (A_i, x_i) is the private key of the i-th user $gsk[i]$.

The group public key is $gpk = (g_1, g_2, u, v, w, h)$ and the group manager secret key is $gmsk = (\xi_1, \xi_2)$ that allows tracing signatures.

Sign Phase

The sign phase generates a signature σ on a message $M \in \{0,1\}^*$. Given as other inputs is a member secret key $gsk[i] = (A_i, x_i)$ and a group public key $gpk = (g_1, g_2, h, u, v, w)$. The signature is computed by the zero-knowledge protocol of the Strong Diffie-Hellman assumption as follows:

1. A signer randomly selects exponents $\alpha, \beta \leftarrow Z_p$ and computes the linear encryption of A represented by values T_1, T_2, T_3,:

$$\begin{aligned} T_1 \;=\; u^{\alpha}, T_2 \;=\; v^{\beta}, T_3 \;=\; A_i h^{\alpha+\beta}, \\ \delta_1 \;=\; \alpha x, \delta_2 \;=\; \beta x. \end{aligned} \tag{4.1}$$

2. The blinding values $r_\alpha, r_\beta, r_x, r_{\delta_1}, r_{\delta_2}$ are randomly picked from Z_p, and values R_1, R_2, R_3, R_4, R_5 are computed:

$$\begin{aligned} R_1 = u^{r_\alpha}, R_2 = v^{r_\beta}, R_3 \;=\; e(T_3, g_2)^{r_x} \cdot e(h, w)^{-r_\alpha - r_\beta} \cdot e(h, g_2)^{-r_{\delta_1} - r_{\delta_2}}, \\ R_4 = T_1^{r_x} u^{-r_{\delta_1}}, R_5 = T_2^{r_x} v^{-r_{\delta_2}}. \end{aligned} \tag{4.2}$$

3. The signer computes a challenge $c \in Z_p$ using the hash function

$$c = H(M, T_1, T_2, T_3, R_1, R_2, R_3, R_4, R_5), \tag{4.3}$$

4. Using the challenge c, values $s_\alpha, s_\beta, s_x, s_{\delta_1}, s_{\delta_2}$ are computed to seal the proof of knowledge of $(\alpha, \beta, x, \delta_1, \delta_2)$

$$\begin{aligned} s_\alpha = r_\alpha + c\alpha, s_\beta = r_\beta + c\beta, s_x = r_x + cx, \\ s_\delta = r_{\delta_1} + c\delta_1, s_{\delta_2} = r_{\delta_2} + c\delta_2. \end{aligned} \tag{4.4}$$

5. Output the signature $\sigma = (T_1, T_2, T_3, R_2, R_3, R_5,\ c, s_\alpha, s_\beta, s_x, s_{\delta_1}, s_{\delta_2})$.

Verification Phase

In the verification phase, a verifier checks the validity of the signature σ generated on the message M by using only the group public key $gpk = (g_1, g_2, h, u, v, w)$.

1. All the values R_1, R_2, R_3, R_4, R_5 are restored:

$$R_1^{'} = u^{s_\alpha} T_1^{-c}, R_2^{'} = v^{s_\beta} T_2^{-c}, R_4^{'} = u^{s-\delta_1} T_1^{s_x}, R_5^{'} = v^{-s_{\delta_2}} T_2^{s_x}, \tag{4.5}$$

$$\begin{aligned} R_3^{'} = e(T_3, g_2)^{s_x} e(h, w)^{(-s_\alpha - s_\beta)} e(h, g_2)^{(-s_{\delta_1} - s_{\delta_2})} \\ (e(T_3, w) e(g_1, g_2)^{-1})^c. \end{aligned} \tag{4.6}$$

2. The verifier restores the challenge $c^{'}$:
   $c^{'} = H(M, T_1, T_2, T_3, R_1^{'}, R_2^{'}, R_3^{'}, R_4^{'}, R_5^{'})$,
   if c is equal with restored c' then the verifier accepts the signature, and rejects otherwise.

Open Phase

An entity who knows the group manager's secret key $gmsk = (\xi_1, \xi_2)$, the group public key $gpk = (g_1, g_2, h, u, v, w)$ together with a message M and the corresponding signature $\sigma = (T_1, T_2, T_3, R_2, R_3, R_5,\ c, s_\alpha, s_\beta, s_x, s_{\delta_1}, s_{\delta_2})$ on this message is able to trace a signer. Firstly, the signature is checked whether is valid for the message. Then, the signer's private parameter A is recovered by $A = T_3/(T_1^{\xi_1} T_2^{\xi_2})$. Finally, A is compared with the elements A_i of the users' private keys, and the index that indicated the signer identity information is looked up.

Security and Signature Length

The BBS04 scheme satisfies these security properties: correctness, full-anonymity and full-traceability. The properties are proved under the random oracle model. The detailed security analysis can be found in [20]. The BBS04 signature consists of three elements of G_1 and six elements of Z_p. Using 170 bit curves ($|G_1| = 171$ bits), the total length of the signature is 1533 bits (192 bytes). The proposed protocol in Chapter 8 is based on the BBS04 group signature schemes.

4.3.3 Pairing-Based Group Signatures with Verifier-Local Revocation

A pairing-based short group signature scheme with a verifier-local revocation is proposed by Boneh and Shacham in [23]. The scheme employs the verifier-local revocation mechanism that uses a revocation list containing users' revocation tokens. In this approach, signers do not need recompute any group signature scheme parameters and keys in the case of the revocation of a user. Only verifiers download and update the revocation list. In the case of breaking the rules, a private key is published online by a trusted authority or a group manager. The revoked users loose their privacy because their signatures signed by their revoked keys become linkable. The security of the BS04 scheme is based on the Strong Diffie-Hellman and the Decisional Linear assumptions. The scheme is secured in the random oracle model and employs two hash functions. The first hash function is a function H mapping $0, 1^* \rightarrow Z_p$. The second hash function is a function H_0 mapping $\{0,1\}^* \rightarrow G_2^2$. This group signature scheme consists of these phases: Key Generation, Sign and Verify (with Revocation Check).

Key Generation Phase

1. A generator $g_2 \in G_2$ is randomly selected and $g_1 = \psi(g_2)$ is set if $e(\psi(g_2), g_1) \neq 1$.
2. A issuer private key is randomly selected $\gamma \leftarrow Z_p^*$ and the third element of a public group key is set as $w = g_2^{\gamma}$. The group public key is $gpk = (g_1, g_2, w)$.
3. For each user (the i-th user of the group), a tuple (A_i, x_i) is computed by the holder of γ (the private-key issuer) such that $A_i = g_1^{\frac{1}{x_i+\gamma}}$ where $x_i \leftarrow Z_p^*$ is randomly selected. The tuple (A_i, x_i) is the private key of the i-th user $gsk[i]$. A revocation token $grt[i] = A_i$ corresponds to the A-element of the i-th user private key.

Sign

The signing phase is performed by a signer who produces a signature σ on a message $M \in \{0,1\}^*$. As other inputs, a member secret key $gsk[i] = (A_i, x_i)$ and a group public key $gpk = (g_1, g_2, w)$ are given. The signature is computed as follows:

1. A signer picks a random nonce $r \leftarrow Z_p$ and obtains generators

$$(\overline{u}, \overline{v}) = H_0(gpk, M, r) \in G_2^2, \tag{4.7}$$

where H_0 is a two-dimensional hash function with mapping $\{0,1\}^*$ to G_2^2, and computes their images in G_1 by

$$u = \psi(\overline{u}), v = \psi(\overline{v}). \tag{4.8}$$

2. The signer randomly selects an exponent $\alpha \in Z_p$ and computes pseudonyms T_1, T_2 by
$$T_1 \ = \ u^{\alpha}, T_2 \ = \ A_i v^{\alpha}. \tag{4.9}$$
3. The signer sets
$$\delta \ = \ \alpha x_i, \tag{4.10}$$
and picks random values $r_\alpha, r_x, r_\delta \in Z_p$.
4. The signer computes helper values by
$$\begin{aligned} R_1 &= u^{r_\alpha}, \\ R_2 = \ e(T_2, g_2)^{r_x} e(v, g_2)^{-r_\delta} \ e(v, w)^{-r_\alpha}&, \\ R_3 &= T_1^{r_x} u^{-r_\delta}. \end{aligned} \tag{4.11}$$
5. The signer computes a challenge c value by
$$c = H(gpk, M, r, T_1, T_2, R_1, R_2, R_3) \in Z_p, \tag{4.12}$$
and response values by
$$\begin{aligned} s_\alpha &= r_\alpha + c\alpha, \\ s_x &= r_x + cx_i, \\ s_\delta &= r_\delta + c\delta. \end{aligned} \tag{4.13}$$
6. The signature is outputted as $\sigma = (r, T_1, T_2, c, s_\alpha, s_x, s_\delta)$.

Verification Phase

The verification phase is performed by a verifier who checks a purported signature σ on a message $M \in \{0,1\}^*$. As other inputs, the group public key $gpk = (g_1, g_2, w)$ and a set RL (Revocation List) of revocation tokens are given. The verify phase consists of two parts, namely Signature check and Revocation check. Both parts are proceeded as follows:

1. A verifier restores generators
$$(\overline{u}, \overline{v}) = H_0(gpk, M, r) \in G_2^2, \tag{4.14}$$
and computes their images in G_1 by
$$u = \psi(\overline{u}), v = \psi(\overline{v}). \tag{4.15}$$

2. The R_1, R_2, R_3 values are restored by

$$\begin{aligned} R_1^{'} &= u^{s_\alpha}/T_1^c, \\ R_2^{'} &= e(T_2, g_2)^{s_x} e(v, g_2)^{-s_\delta}\ e(v, w)^{-s_\alpha} \cdot (e(T_2, w)/e(g_1, g_2))^c, \\ R_3^{'} &= T_1^{s_x} u^{-s_\delta}. \end{aligned} \tag{4.16}$$

3. The verifier restores the challenge $c^{'}$:
   $c^{'} = H(gpk, M, r, T_1, T_2, R_1^{'}, R_2^{'}, R_3^{'})$,
   if a purported c is equal with c' restored then the verifier accepts the signature, and rejects otherwise.
4. The verifier proceeds Revocation check. Each element $A \in RL$ is checked whether is encoded in (T_1, T_2) by

$$e(T_2/A, \overline{u}) = e(T_1, \overline{v}). \tag{4.17}$$

   If no element of RL is encoded in (T_1, T_2), the signer of the signature has not been revoked. In revocation check, the verifier has to check all A elements in the revocation list with purported elements (T_1, T_2) and restored elements $\overline{u}$ and $\overline{v}$.

Security and Signature Length

The BS04 scheme satisfies these security properties: correctness, selfless-anonymity and traceability. The properties are proved under the random oracle model. The detailed security analysis can be found in [23]. The BS04 signature consists of two elements of G_1 and five elements of Z_p. Using 170 bit curves ($|G_1| = 171$ bits), the total length of the signature is 1192 bits (149 bytes). The proposed protocol in Chapter 7 is based on the BS04 group signature schemes.

5 STATE OF THE ART

This chapter summarizes the state of the art of privacy-preserving cryptographic protocols which are employed to secure communication in heterogeneous networks. Due to a huge number of cryptographic protocols, this analysis is focused on the cryptographic schemes that secure communication between nodes where data integrity, authentication, non-repudiation, privacy and session unlinkability are required. These properties are usually ensured by digital signatures, authentication schemes and group signature schemes. These security approaches can be designed to secure communication at different layers (link, network, transport, application). In fact, anonymous and security solutions in heterogeneous networks are mainly focused on the network layer where the IP protocol is common for various technologies and networks.

Nevertheless, the proposed protocols in the thesis are mainly designed for certain applications in heterogeneous networks such as VANET (Chapter 8) and geolocation (Chapter 9). These applications are mostly secured by specific cryptographic solutions at the application layer. Furthermore, the proposed protocols in Chapters 7, 8, 9 are based on group signature schemes. Due to this fact, the evolution and evaluation of group signature schemes are analyzed as well.

5.1 Anonymous and Security Solutions in Heterogeneous Networks

Recent years have witnessed a number of privacy-enhancing and security solutions in heterogeneous networks. Researchers focus mainly on wireless networks, namely mesh networks and ad hoc networks. The group of wireless networks such as the Mobile Ad hoc Network (MANET), Vehicular Ad hoc Network (VANET), Wireless Sensor Network (WSN) and Wireless Mesh Network (WMN) can be called as Self-Organizing Networks (SONs), and the basic security issues of these networks are summarized in [137]. Nevertheless, many proposals and security solutions are focused only on the concrete network technology and topology. In fact, lot of these solutions such as anonymous routing schemes and secure roaming schemes solve the security and privacy at the network layer. In the following subsections, these security and privacy-enhancing solutions for MANET, WMN and hybrid wireless networks are discussed.

5.1.1 Anonynous Routing in Mobile Ad hoc Networks

Several studies and papers deal with anonymous communications in MANETs. Many of them focus on anonymous routing protocols and key management.

Zhang *et al.* [180] propose an anonymous communication protocol for Mobile Ad hoc Networks (MANETs), termed MASK. The protocol uses dynamic changing pseudonyms to hide identities of senders from outside observers. MASK uses a pair-based authentication protocol between neighbors which ensures anonymity of senders, receivers and sender-receiver relationships. Only a trusted third party can link pseudonyms and reveal nodes' identities in MASK.

Huang [84] deals with anonymous communication in mobile ad hoc networks and in wireless broadcasting environments. The solution is based on identity-based cryptography and uses pseudonyms and blind signatures to set up anonymous communication sessions. The proposed solution provides a pseudonym-based encryption and the revocation of pseudonyms. However, the solution has few flaws such as the presence of a centralized party called Private Key Generator (PKG) which is a trusted third party, and the vulnerability of the pseudonym-based encryption scheme to the Sybil attack (i.e. forging identities/pseudonyms) because of any device with an initial identifier can generate an arbitrary number of pseudonyms [125].

Seys and Preneel [153] describe an Anonymous on demand Routing protocol for MANETs (ARM) based on one-time public/private key pairs to provide destination privacy. In ARM, each authorized node (a source) pre-shares a unique symmetric key with its neighbors (1-hop destination) to encrypt route reply messages.

El Defrawy and Tsudik [64] introduce a PRISM scheme that is on-demand anonymous MANET routing protocol based on group signatures. In PRISM, nodes have no a priori topology knowledge and have to first determine their geographical area of interest and probe it with a route-request message (RREQ). Later, these authors present the ALARM scheme [63]. ALARM is also based on group signatures which construct one-time pseudonyms to identify nodes at their current locations. ALARM is a link-state protocol. In ALARM, nodes know the entire MANET topology before their communication. Hence, precise destination addressing is used. On the other hand, the privacy decreases by exposing the topology information.

More information about security and privacy in MANETs and overviews of anonymous routing protocols in MANETs can be found in papers [44], [94] and [130].

5.1.2 Security and Privacy in Wireless Mesh Networks

Furthermore, there are proposals enhancing the privacy and security in Wireless Mesh Networks (WMN). Sun *et al.* [158] propose a security architecture to ensure

unconditional anonymity for honest users and the traceability of misbehaving users in wireless mesh networks. The solution uses tickets, self-generated pseudonyms and a hierarchical identity-based cryptographic scheme. The solution uses a blind signature technique and pseudonyms to achieve user privacy by delinking user identities from their accesses. The pseudonym generation mechanism does not rely on a central authority but the system does not provide routing anonymity.

Wan *et al.* [167] present two solutions for security and privacy protection in Wireless Metropolitan Mesh Networks (WMMN). In the first solution, group signatures (BBS04) are used to anonymously establish session keys and enforce access control. In this solution, the user's identities are protected from eavesdroppers but have to be disclosed to mesh router because of routing in the mesh backbone. The second solution uses pairwise shared secrets along with group signatures to keep mesh clients anonymous from mesh routers.

Sgora *et al.* [154] provide the overview of security and privacy issues in wireless mesh networks. The paper discusses the proposals focusing on intrusion prevention mechanisms, security routing and intrusion detection systems. Another overview is provided by Sen [152] who focuses more on privacy issues.

5.1.3 Secure and Anonymous Roaming Authentication Protocols in Wireless Networks

Anonymous roaming authentication has been addressed by several solutions, e.g. [170], [173], [81]. These solution usually provide secure and privacy-enhancing authentication in roaming services. These services allow legal users to get access to wireless network services in foreign domains.

Yang *et al.* [173] propose two secure roaming protocols which can be used in various kinds of roaming networks such as cellular networks and interconnected wireless local area networks. The first protocol is a two-party authentication protocol with strong user anonymity based on group signatures (BS04). The second protocol is a two-party authentication protocol with weak anonymity and is based on identity based signatures. Both protocols are universal and provide key establishment by a challenge-response approach. Nevertheless, in the paper [81], the authors point out that the both protocols have some flaws. The protocols do not provide a DoS attack resistance. An adversary can send a large volume of forged login requests to exhaust the storage and processing resources of servers. Secondly, the protocols do not provide user untraceablility and backward and forward unlinkabilities because a foreign server is able to identify all protocol runs where the user has and will be involved.

He *et al.* [81] propose a privacy-preserving universal authentication protocol for wireless communications. The two-party authentication and key agreement protocol is based on group signatures (BS04) which performs 4 pairing operations and 15.75 elliptic curve scalar multiplication on a roaming user. Nevertheless, the number of operation in the revocation and a revocation list size increase with the number of revoked users and with the number of user secret keys. More secret keys provide backward unlinkability. Nevertheless, the checking operation can be misused by an adversary to launch a resource depletion attack on the foreign servers.

Wen *et al.* [170] propose a smart card-based anonymous user authentication scheme for global roaming services in Global Mobility Networks (GLOMONET). This scheme provides user anonymity, mutual two-factor authentication, key agreement, Denial of Service (DoS) resistance and replay attack resistance. However, the scheme efficiently solves authentication and key establishment but there is assumption that every user has a smart card. Moreover, the scheme does not offer the anonymous communication after the authentication phase.

The similar work [89] proposes a three-round anonymous roaming protocol. The proposed protocol does not require the participation of home servers in wireless mobile networks. The protocol uses a pseudo-identity-based signcryption scheme to perform efficient revocation with a short revocation list. Signcryption minimizes the number of pseudo-identities which are stored in a Subscriber Identification Module (SIM) card. However, this protocol are solely designed for mobile networks with SIM cards.

5.1.4 Security and Privacy in Hybrid Wireless Networks

The application of schemes mentioned in the previous subsections can be hard on heterogeneous networks that aggregate more network technologies. In this case, a cryptographic solution or framework that runs on application layer is needed.

Prasad *et al.* [141] discuss Authentication, Authorization and Accounting (AAA) services and end-to-end security in heterogeneous access networks. In the paper, the authors present the security objectives for mobility on IP networks. The authors discuss existing AAA protocols such as DIAMETER and RADIUS, however, the privacy-enhancing solutions are not considered.

Capkun *et al.* [42] propose a secure and privacy-preserving communication protocol in hybrid ad hoc networks. The protocol offers secure communication and protects a user anonymity and a location privacy. The proposed approach is based on frequently changing node pseudonyms and cryptographic keys. Each node stores a set of public/private key pairs and certificates with different pseudonyms signed

by a trusted third party. The node uses a key pair during the authentication process and during the establishment of shared symmetric keys with neighbors. To ensure the privacy, node's public/private key pair and shared symmetric keys with its neighbors are periodically changed. Nevertheless, this approach has several flaws. The revocation of keys is problematic due to using a large number of keys in the network. Then, periodically refilling the keys may burden the network. Moreover, each node with a set of public/private keys and certificates requires a large storage space.

The paper [4] presents an anonymous communication protocol that preserves (user,server)- anonymity in mobile hybrid networks that involves cellular, wired and wireless (WiFi) connections. The communication anonymity is provided without assuming trusted mobile network operators and the approach is designed for an anonymous communication based on a request/response messages such as web browsing, social network activities, posts in blogs, small-file uploads and so on. However, this approach does not provide the authentication of users, non-repudiation and does not ensure a user responsibility.

Mahmoud *et al.* [113] propose a lightweight secure and privacy-preserving protocol for hybrid ad hoc wireless network. The scheme is based on short-life pseudonyms, one-time session keys, and per-hop encryption/decryption operations to preserve users' privacy. Nevertheless, this scheme is suitable only for multihop packet relay services. Then, the scheme is not proper for services that do not need to know the exact locations of source nodes such as in geolocation and VANET applications.

In summary, there are many solutions which offer some privacy-preserving methods. Nevertheless, these solutions focus often on one technology or services such as routing, roaming, mesh communication and so on. On the other hand, only few solutions (mentioned in the previous subsections) provide secure and private communication in hybrid networks. But these solutions have flaws which are described in these previous subsections.

5.2 Security and Privacy in Vehicular Ad hoc Networks

The protocol proposed in Chapter 8 provides a secure and privacy-friendly solution for the vehicular networks that are consisted of several network technologies. Due to this fact, recent privacy and security solutions of VANETs are discussed in the following text.

Privacy in VANETs can be achieved in many ways. For example, in the paper [50] the authors deal with privacy and security in VANETs with a safe distance-

based location privacy scheme called SafeAnon. The scheme uses a safe distance measurement technique to determine the maximum obfuscation radius for preserving location privacy while maintaining traffic safety. The SafeAnon scheme fights against a Global Passive Adversary (GPA) that can locate and track any vehicle in an area of interest by eavesdropping on broadcast messages. Nevertheless, this protection can be only employed in several VANET applications based on a short distance among vehicles, e.g., collision detection. In comparison with this scheme, the proposed protocol in Chapter 8 aims at VANET applications used in medium and long distances, e.g., the detection of traffic jams, accidents and so on. Horng *et al.* [82] propose a private V2V communication mode that can be used in wide areas. Nevertheless, the main drawback of these proposed private V2V scheme is the restriction of privacy which can be kept only in the specific group of users, where users know the public keys of other participants and can build their profiles. The second disadvantage is the presence of a session key establishment subphase which can slow the communication process.

Using pseudonyms in VANETs is proposed in [72] and [69]. Raya and Hubaux [144] use anonymous certificates which are stored in vehicles (usually in a tamper-proof device). This approach uses a set of short-lived pseudonyms, and privacy among vehicles is provided by changing these certified public keys. Nevertheless, in large urban VANETs, this approach is burdened by preloading and storing a large number of anonymous certificates with pseudonyms.

To provide privacy and security in VANETs, the solutions may use group signatures. Group signatures (GS) provide user anonymity by signing a message on behalf of a group. GS guarantee the unlinkability of honest users and the traceability of misbehaving users. VANET security solutions based on group signatures are analyzed in the following text. The scheme [108], called GSIS, uses the combination of a group signature scheme (BBS04) [20] with a hybrid membership revocation mechanism in the V2V communication, and Identity Based Group Signature (IBGS) in the V2I communication. The hybrid membership revocation with the list of revoked members (RL) works with a threshold value T_τ. In case $|\mathrm{RL}| < T_\tau$, the scheme uses a revocation verification algorithm. Otherwise, the scheme updates the public/private group keys of all non-revoked members. For efficient verification, the authors of the paper [177] propose a GS with batch verification in V2I, which takes three pairing operations. This scheme, called IBV, has several drawbacks such as using tamper proof devices, being thus vulnerable to tracking or impersonation attacks. The complete description can be found in [52]. Schemes proposed in [179] and [168] can efficiently verify a large number of messages in V2V. These schemes use short group signatures with fast batch verification (only two pairing operations are used instead of 5 n, where n is the number of messages). Nevertheless, the per-

formance of batch verification degrades in dense V2V communication with bogus messages. The On Board Units (OBUs) must process the messages quickly (they have between 100 ms and 300 ms to process a message [86]). Thus, the computation of expensive pairing and exponentiation on limited On Board Units (OBUs) is a hard requirement to meet because of the short response time. This fact limits the security of VANETs in practice. Qin *et al.* [143] employ an identity-based group signature scheme with the batch verification, provides a scalable management of large VANETs and an efficient revocation of members, but suffers from more expensive signing and verification phases than phases in solutions based on group signatures.

Generally, related VANET security solutions usually deal with two problems. Firstly, the solutions based on pseudonyms can be fast, but using the many pseudonyms burdens the communication and management of VANET systems. Secondly, the solutions based on group signatures employ only one public key to verify signed messages, and provide user privacy and unlinkability. Nevertheless, these solutions have problems with efficiency (many expensive operations) during signing or verification or with DDoS attacks. Protocol 2 (Chapter 8) focuses on these problems.

5.3 Group Signatures Schemes

The proposed protocols in Chapters 8 and 7 are based on group signatures. This section discusses the evolution of group signature schemes and evaluates these schemes.

5.3.1 Evolution of Group Signature Schemes

Group signatures were introduced and first four schemes were presented by Chaum and Heyst [47] in 1991. The main disadvantage of these schemes is long sizes of a group public key gpk and a signature. These sizes depend on the number of members in a group. If a new member is added to the group, it is necessary to modify gpk. These deficiencies are very impractical for large groups of members. Therefore, these schemes are not suitable for many services and applications with a large number of users. In the work CS97 [39], published in 1997, authors propose a scheme which uses the constant size of gpk and signatures. New members can be added to the group without the need to generate a new key pair gpk and group member secret key $gsk[i]$. The paper ACJT00 [7], introduced in 2000, presents an efficient scheme which is resistant of coalition, i.e. it is impossible for a subset of group members including a group manager to create a valid signature. The disadvantage of the scheme is missing of the revocation of group members and prevention to a revoked member generates the valid signatures on behalf of the group. The work AST02 [8], published in 2002, is based on the scheme ACJT00

[7] and adds the revocation of the group members without using a time stamp. This approach keeps a constant length of a signature, i.e. this length does not increase linearly with the number of revoked members. However, the scheme has more operations in signing and verification phases than related schemes. The scheme TX03 [165], published in 2003, provides the dynamic revocation of group members. Revoked members are no longer able to create a valid signature. On the other hand, the disadvantage is that, *gpk* has to be recalculated when a member is added to the group or remove from the group. This approach is highly inefficient in the real time systems working with large groups. The schemes BS04 [23] and BBS04 [20], published in 2004, allow creating short group signatures. These schemes are based on bilinear maps and produce short signatures which are suitable in systems where bandwidth is restricted. Unless as the previous schemes that are secure in the random oracle model, the scheme BMW03 [15], introduced in 2003, is secure in the standard model. Nevertheless, this scheme is designed for the static and small groups of users. Therefore, this scheme is not proper for services with a large number of users.

The scheme ACHM05 [6], introduced in 2005, is provably secure in the standard model and works with dynamic groups. The scheme provides anonymity, unforgeability, untraceability and exculpability, and is secure against a non-adaptive adversary who does not have $gsk[i]$ of group members. Nevertheless, the scheme does not achieve forward anonymity and security is proved under non-standard assumptions.

The scheme DP06 [57], proposed in 2006, ensures the security in the RO-model and works with dynamic user groups. This scheme provides the complete anonymity of members' signatures with very short sizes (i.e. around 181 B), shorter than the scheme BBS04.

The scheme BW06 [25] provides the provable security in the standard model. But, the size of the signature depends on the size of the group. The newer scheme BW07 [26], introduced in 2007, produces shorter and almost constantly sized signature in comparison with the previous schemes. The length of a signature increases logarithmically as the size of the group.

The scheme LCSL07 [102] produces short signatures with constant lengths. This scheme offers full anonymity and full traceability, and the public key and signatures are shorter than in the previous schemes (5 group elements). Nevertheless, the verify algorithm takes 6 pairings.

The scheme G07 [76], published in 2007, ensures full anonymity in the standard model. The scheme is based on bilinear groups and produces the constant lengths of keys and signatures. The scheme also supports the dynamic addition of new members to the group. Anyway, the signature consists of about 50 elements.

The scheme LCHH09 [100] provides a reversible user revocation. If a user is

revoked by a mistake and a group manager make a decision that the user can remain in a group after this event then the user can keep his/her group user secret key without any new distribution process.

The scheme BCNSW10 [19] allows creating even shorter signatures than the previous schemes and the security of the scheme is provided in the RO model.

In 2011, Hwang et al. [87] proposed another pairing-based short group signatures with controllable linkability. The scheme is based on the q-Strong Diffie-Hellman problem and the revocation is solved by a key update approach. The scheme is similar like the BBS04 scheme [20].

In 2012, Libert et al. [104] proposed a scalable revocable group signature scheme. This scheme ensures the security in the standard model. Nevertheless, the scheme produces large signatures (96 group elements, i.e. around 6 kB.) that is not convenient for practical deployment in systems with restricted bandwidth. The scheme is not proper for large systems due to the many operations in the signing and verification phases that depend on the number of users.

Several group signature schemes employ batch verification to get more efficient verification of signed messages. The efficiency of batch verification is studied in the paper [66]. The verification of n messages during one batch can reduce the number of expensive bilinear pairing operations. More information about batch verification can be found in Chapter 6.

The group signatures schemes have to deal with the revocation of dishonest users, attackers or users who left group. There are three ways how to revoke user from the group. The first and naive method is based on the reinitialization of group public key and sending it to all unrevoked members who must re-computate their group member secret keys. The second method is based on an accumulator, and the third method employs a list with revoked users. These revocation methods are more described in Chapter 7.

Besides the problems with revocation, group signature schemes cause problems with their implementations on memory and computational restricted devices due to more expensive operations and larger signatures or keys in comparing with classic signature schemes such as RSA, ECDSA and Message Authentication Code (MAC) schemes. Also an efficient verification in short time and/or the verification of many signatures can be the bottleneck of these cryptographic schemes.

5.3.2 Evaluation of Group Signature Schemes

Group signature schemes can be evaluated by their performance, signature sizes, parameter sizes and security assumptions used. The performance of group signature schemes strongly depends on two main phases: signing and verification. More

efficient schemes in the signing phase are convenient for different applications than the efficient schemes in the verification phase. The parameters of chosen group signature schemes are compared in Table 5.1. The operations and marks are abbreviated as p - bilinear pairings, e - exponentiation, mul - multiplication, div - division, add - addition (subtraction), H - hash, k - the length of identities in bits, m - the length of message in bits, RL - the number of members in a revocation list, EF - efficiently computable isomorphism from G2 to G1, T - the total time of a period, N - number of members. In Table 5.1 the performance of several schemes, e.i. the number of signing operations, the number of verification operations or the sizes of signatures are usually outlined without any optimization and improvements. Thus, some schemes can be optimized, for example, schemes BBS04, DP08 and HLCCN enable precomputing all pairing operations in the signing.

Usually, the schemes which are secured in the random oracle model are more efficient in signing and verification phases and have shorter signatures than schemes secured in the standard model. It is not easy to determine the number of operations in the phases of schemes which are secured in the Standard model due to these schemes use various sets of vectors and parameters that causes many arithmetic operations. For example, the scheme G07 [76] consists in the signing algorithm of a Non-Interactive Witness-Indistinguishable (NIWI) proof of knowledge, the Boneh-Boyen signature scheme, a public key encryption and a Non-Interactive Zero Knowledge (NIZK) proof. On the other hand, the scheme BCNSW10 [19] is the most efficient from schemes compared in the signature size, i.e. only 3 group elements in G_1 and 2 elements in Z_q. The most efficient scheme in the signing phase is the NS04 scheme [133] due to the scheme has 0 pairing operation and only 3 exponentiation operations. The most efficient scheme in the verification phase is the scheme BCNSW10 [19] due to 2 pairing and 3 exponentiation operations.

In summary, group signatures can have three bottlenecks that are a user revocation, the efficiency of phases (signing, verification) and the length of signatures. In this thesis, the schemes BBS04 [20] and BS04 [23] are chosen due to their balanced properties and they are further optimized and modified to fit in the proposed group signature-based cryptographic protocols for the chosen communication services in heterogeneous networks. Protocol 1 (Chapter 7) aims at the user revocation. Protocol 2 (Chapter 8) deals with the efficiency of phases (signing, verification) and a DDoS mitigation in the verification phase.

Tab. 5.1: Parameters of Group Signatures.

Scheme	**Signing operations**	**Verification operations**	**Size of signature**	**Size of group public key**	**Efficiency**	**Security model**	**Type**
ACJT00 [7]	14e + 1H+ 9mul + 2div + 6add	15e + 1H + 9mul + 2div + 4sum	8696 b	6144 b	Constant gpk and sign.	Random Oracle	Non-bilinear
NS04 [133]	3e + 32mul + 14add + 1H	3p + 2e + 14mul + 8add + 1H	4776 b	2904 b	Constant gpk and sign.	Random Oracle	Bilinear
BBS04 [20]	3p + 12e + 10mul + 8add + 1H	5p+12e +7mul + 1div + 2add + 1H	3|G1| + 6|Zp| 1 553 b	6|G1| 1026 b	Constant gpk and sign.	Random Oracle	Bilinear
BS04 [23]	3p + 2EF + 8e +8mul + 3add + 2H	(6 +|RL|)p + 8e + 4mul + 2div +2H + 2EF	2|G1| + 5|Zp| 1 192 b	3|G1| 513 b	Constant gpk and sign.	Random Oracle	Bilinear
ACHM05 [6]	12e + 2div + 3add	10p + 1e + 3mul	6|G1|+2|G2| 2052 b	2|G1| + 4|G2| + |GT|	Constant gpk and sign.	Interactive assumptions avoiding random oracle	Bilinear
BW06 [25]	(5k + m + 5)e + (4k + m + 4)mul + (2k-1)add	(3+2k)p + me + (m+k)mul	(2k + 3)|G|	(k + m + 3)|G| + |Gq| + |GT|	Logarithmic gpk and sign.	Standard	Bilinear
ZL06 [181]	2p + 17e + 17mul + 7add + 2div + 1H	(3 + RL)p +17e +9mul + 2div + 1H	8|Zp| + 5|G| 2215 b	(3 + T)|G|	Constant sign.	Random oracle	Bilinear
DP06 [57]	3p + 3e + 15mul + 7add + 1div + 1H	5p +4e +11mul + 3add + 2div	4|Zq| + 4|G| + 1|H| (1444 b)	4|G1|+2|G2| +3|GT|	Constant sign.	Random oracle	Bilinear
BW07 [26]	(12 + 2m)e + (11 + 2m)mul	6p +(3 + m)e + (4 + m)mul	6|G| 1026 b	(4 + m)|G| + |Gq| + |GT|	Logarithmic gpk a constant sign.	Standard	Bilinear
LCSL07 [102]	12e + 10mul + 1div + 1H + 1add	6p + 3e + 4mul	5|G|	3|G| + |Gq|	Constant gpk and sign.	Standard	Bilinear
G07 [76]	NIWI + NIZK + BB signature	246p ([62])	50|G|	8|G|	Constant sign.	Standard	Bilinear
BCNSW10 [19]	4p + 5e + 2m + 1div + 1add + 1h	2p + 3e + 1m + 1div	3|G1| + 2|Zq|	2|G|	Constant gpk and sign.	Random Oracle	Bilinear
HLCCN11 [87]	4p + 11e + 11m + 5add	6p + 10e + 7m + 1div + 1h	3|G1| + 5|Zq| (1363 b)	6|G1| + 2|G2| + 4|GT|	Constant gpk and sign.	Random Oracle	Bilinear
LPY12 [104]	(logN)e	4p + 2e + the verification of one time signature	96|G| (6kB)	|G| + |Gq|	Constant sign. and logarithmic gsk and gpk	Standard	Bilinear

6 PERFORMANCE ANALYSIS OF CRYPTOGRAPHIC PRIMITIVES AND OPTIMIZATION TECHNIQUES

The goal of this chapter is to evaluate the performance of cryptographic primitives and figure out security parameters for devices used in heterogeneous networks. The chapter also presents optimization techniques that reduce the number of operations in cryptographic protocols. Further, it is focused on the optimization techniques that can be used in digital signatures, group signature schemes and pairing-based schemes.

6.1 Performance Analysis of Cryptographic Primitives and Modular Arithmetic

Cryptographic primitives and math operations used in cryptography have different computational and memory requirements. Hand-held and embedded devices usually offer sufficient computational and memory performance also for asymmetric cryptographic primitives and more expensive math operations such as exponentiations and bilinear pairings. Unlike smart-cards and some highly-restricted microcontrollers which have lower computational and memory performance, hand-held and embedded devices allow employing the advanced cryptographic and math libraries. These libraries offer optimized cryptographic primitives and operations and enable designing the privacy-preserving cryptography solutions. Moreover, to speed up some cryptographic operations, the most widespread devices often have some cryptographic support provided by a dedicated chip. How to use this chip for increasing the performance is described in subsection 6.1.3.

6.1.1 Performance Results of Cryptographic Operations

The cryptographic components and modular arithmetic operations can be implemented in many languages and on many platforms. To compare cryptographic operations with pairing-based operations, the object-oriented programming languages (JAVA, C#, C++) have been used to obtain the following results.

Performance Results of Cryptographic Operations on PC

Current PC machines provide sufficient computational and memory performance for various cryptographic primitives. Table 6.1 shows the average times of crypto-

graphic operations measured in the JAVA implementation. The results are measured on a PC machine with Intel(R) Xeon(R) CPU X3440 @ 2.53GHz, 4 GB Ram, Windows 7 Professional. The average values are taken from 1000 measurements executed on the machine used. The modular arithmetic operations are provided by the java.math.BigInteger class. The pairing based operations are implemented by the Java Pairing Based Cryptography (jPBC) Library [1]. The implementation employs the MNT curves type D with the embedding degree $k = 6$, the 171-bit order of curves and the pre-generated parameters d840347-175-161.param. The schemes based on RSA employ the size of a group element of 1024 bit. The pairing operation takes about 40 ms and is the most expensive operation measured. The hash operation applied on curve elements takes about 1 ms due to need to map a hash value to a curve element. Addition, multiplication and the hash of RSA integers are more efficient operations (up to tens μs) than exponentiation which takes approx. 5 ms.

Tab. 6.1: Times of Cryptographic Operations on PC machine.

Elements:	1024-bit RSA based	170-bit Curve Based
Operation:	Average Time [μs]	
Pairing	-	40640
Exponentiation	5370	5160
Multiplication	28	13
Hash (SHA-1)	16	1050 (with map to a curve)
Addition (Subtraction)	5	5

6.1.2 Performance Results of Modular Arithmetic and Selected Cryptographic Primitives on Constrained Devices

Modular arithmetic operations with large numbers and moduli can be a computational bottleneck on resource restricted devices like smart-cards, hand-held devices or sensors, especially, multiplication and exponentiation operations. The modular arithmetic is important for a wide variety of computations in these devices, from communication to signal processing. Many asymmetric protocols used today for authentication and digital signatures work with multiplicative groups modulo large numbers. The most common examples are the RSA (Rivest, Shamir, Adleman) [147] or DSA (Digital Signature Algorithm) [97] algorithms. But, there are more

[1] (available on http://gas.dia.unisa.it/projects/jpbc/index.html)

cryptographic primitives that can be based on modular operations, such as the Zero-Knowledge protocols [73], Proofs of Knowledge (PK) schemes, e.g. Schnorr's scheme [149], and so on. From the knowledge of the construction of these advanced protocols and the knowledge of performance of underlying operations, the performance of these protocols can be predicted. Based on the implementation of atomic operations on various devices, it is possible to estimate the theoretical performance of the selected primitives such as commitment schemes, the proof of knowledge schemes and so on.

It can be observed from the cryptographic background in Section 4 that the proof of knowledge protocols heavily rely on arithmetic operations in groups where the discrete logarithm operation is hard to compute. Namely, modular operations with moduli in orders of thousand bits are required. These operations are usually available on the PC platform in the form of BigInt libraries (such as OpenSSL, Bouncy Castle, etc.). Unfortunately, these libraries are missing on smart-cards. Only the MultOS platform supports direct modular operations. Thus, the following operations have to be implemented on many platforms and devices without the support of existing libraries. The bit-lengths of moduli and operands are selected according to the most popular group sizes in cryptography (1024 and 2048 bit modulus). Additionally to modular operations, some non-modular (plain) big-integer operations are implemented as they are contained in PK protocols which operate in hidden order groups (1024 bit groups and 160 bit secrets such as w). The results are outlined in Table 6.2.

The measured operations are denoted as follows:

- **MPow**[***mod***] [***exp***]: Modular Exponentiation with *mod* b modulus and *exp* b exponent.
- **MMul** [***len***]: Modular Multiplication with *len* b modulus and operands.
- **Mult320**: Multiplication of two 320 b numbers.
- **Sub400**: Subtraction of two 400 b numbers.
- **RNG**: The random number generation of a 560 b number.
- **PK**: Proofs of Knowledge of discrete logarithm (Schnorr's protocol), $|w| = 160$ b, $|g|$ is a generator in Z_q where $|q| = 160$ b, $|c| = 1024$ b.

Using these benchmarks, it is possible to easily predict the approximate time of newly designed protocols or cryptographic schemes. More results can be found in the paper [79].

Tab. 6.2: Performance Estimation Based on Benchmarks.

	Time in milliseconds							
	S1	**S2**	**S3**	**S4**	**S5**	**A1**	**A2**	**A3**
MPow2048 560b	1047	1273	844	-	539	64.35	42.66	41.11
MPow2048 160b	958	673	440	-	386	20.19	13.32	12.96
MPow1024 368b	204	664	214	90	254	14.83	9.69	8.83
MPow1024 160b	186	477	166	58	226	6.13	4.30	3.97
MMul 2048b	448	392	691	51	29	0.25	0.20	0.20
MMul 1024b	205	187	353	37	28	0.16	0.14	0.10
Mul 320b	101	106	133	34	29	0.03	0.03	0.02
Sub 400b	15	9	33	33	27	0.01	0.02	0.01
Hash SHA1 20kB	111	38	155	381	92	0.12	0.02	0,02
RNG 560 b	5	52	53	65	42	0.12	0,08	0.08
$c = g^w$ (DL commitment)	186	476	165	226	58	6	4	4
$c = g^w h^r$ (Pedersen commit.)	580	1161	717	513	195	12	9	8
$PK\{w : c = g^w\}$	325	830	433	352	222	15	10	9

Glossary:

S1: Oberthur Technologies ID-One Cosmo V7.0-A

S2: Gemalto TOP IM GX4

S3: Gemalto .NET V2+

S4: MultOS ML2-80K-65

S5: MultOS ML3-36K-R1

A1: Samsung Galaxy S i9000 (smart-phone)

A2: Samsung Galaxy Nexus I9250M (smart-phone)

A3: ASUS TF 300T (tablet)

Performance Results of Pairing-Based Cryptographic Operations on Hand-held Devices

It can be expected that some expensive operations such as pairings are more time consuming on less computational powerful devices, e.g., smartphones. The times of bilinear pairing operations and other operations used in pairing-based cryptographic schemes are measured on two smartphones: Samsung Nexus i9250 (OS: Android v4.2.2, RAM: 1 GB, CPU: ARMv7-A Dual-core 1.2 GHz Cortex-A9) and LG Nexus 5 (OS: Android v4.4, RAM: 2 GB, CPU: ARMv7 Quad-core 2.3 GHz Krait 400). Nexus 5 enables using both virtual machines, Dalvik and ART. The results are outlined in Table 6.3. The average values are taken from 10 measurements executed on smartphones. The pairing operation is the pairing type **d**(175) with parameters d840347-175-161. This asymmetric **d** pairing takes 3597.03 ms on Nexus i9250. Nevertheless, the pairing takes only 2383.3 ms on Nexus 5 with ART, or 3016.9 ms on Nexus 5 with Dalvik. In this case, ART outperforms Dalvik by ca. 21%. Exponentiation PowZn u^{α} (both elements are in G_1) is faster than pairing and takes ca 87 ms on Nexus 5 with ART. The hash operation with mapping to Z_n field, which is applied on curve elements, takes 0.20 ms on Nexus 5 with ART. Multiplication of two elements in G_1 (MulZn) takes ca. 0.05 ms on all the smartphones used. In the measurement, the addition and other relatively fast operations are ommited. More results can be found in the paper [122].

Tab. 6.3: Performance of Pairing-Based Cryptographic Operations on Hand-held Devices.

Devices:	Nexus i9250	Nexus 5 (Dalvik)	Nexus 5 (ART)
Operation:	Average Time [ms]		
Pairing e	3597.03	3016.91	2383.30
Exponentiation PowZn	131.73	105.82	87.87
Multiplication MulZn	0.06	0.06	0.05
Hash with map to Zn H()	1.26	0.55	0.20

6.1.3 Efficient Modular Multiplication by Using Coprocessor

In the journal paper [118], the comparison of the accelerated method of multiplication with three classical methods for (modular) multiplication are provided. The standard methods are represented by the operand-scanning multiplication algorithm (the schoolbook method), the product-scanning method (Comba's method) and the Montgomery multiplication. The accelerated method is based on using the RSA

encryption support for multiplication tunneling. The goal is to use the resources of a crypto-coprocessor to accelerate general modular operations such as exponentiation by two.

The search for a fast modular multiplication algorithm is motivated by its need in the attribute authentication systems which need modular arithmetic operations like subtraction, addition, multiplication and exponentiations to be efficiently computed by the restricted devices. Since the exponentiation operation can be done directly by the RSA function (which is simply an exponentiation), the most demanding modular operation is the multiplication. The computation complexity of multiplication is $O(n^2)$, where n is the length of input operands, in most cases equal to the length of the modulus. Fortunately, a trick with RSA method [147] can be used to reduce the computation complexity to almost a constant value given by the RSA implementation. This trick is called RSA tunnel method.

RSA Tunnel Method

RSA algorithm introduced in [147] is the most common algorithm for public-key cryptography. The RSA function is a simple exponentiation (1),

$$c = m^e \mod n \tag{6.1}$$

where m is a message for encryption, e is a public key, n is a public modulus and the result is c, the cipher text. By a wise choice of RSA encryption parameters m, e, n the RSA function can be used for the exponentiation and multiplication. However, the RSA tunnel works only on devices having an accelerated RSA support (without a dedicated chip for the RSA acceleration there would be no performance gain).

Exponentiation is provided directly by importing parameters where m is the base, e is the power and n is the modulus of the intended exponentiation operation. For example, the cryptographic API of the .NET smart-card allows setting the exponent e and the modulus n directly. Then it is necessary to call the RSA method with the base as a parameter and choose no padding for it. The RSA encryption (1) gives the output in the form of a byte array.

The modular multiplication operation is not so straightforward, nevertheless it is also very simple. To use the performance of the crypto co-processor with RSA for the modular multiplication of integers a and b, the binomial formula (2) is used.

$$ab = \frac{(a+b)^2 - a^2 - b^2}{2} \tag{6.2}$$

By such a conversion, a single multiplication is transformed to three exponentiations, three additions and one division by 2 (just a bit shift). But since the addition is efficient and the exponentiation can use the RSA function, all resulting operations

are much faster than a single multiplication. The larger the input parameters are, the more efficient the transformation is. The computational complexity of the RSA Tunnel method depends mostly on the length of the modulus n and grows much slower than in previous methods.

Results of Modular Multiplication Methods

The following results are measured on .NET smart-card (.NET Gemalto V2+, 66 MHz CPU, 16 kB RAM). The time of addition, subtraction and exponentiation operations with large integers is satisfactory due to low complexity (or the direct use of RSA in case of exponentiation). All these operations can be efficiently computed in tens of milliseconds on the chosen .NET smart-card. Moreover, the actual time of these simple operations is negligible in comparison to the card initialization time (cca. 150 ms for card initialization, 120 ms for communication).

On the other hand, multiplication is more challenging because it has the computational complexity $O(n^2)$. The schoolbook and Comba's methods depend on the length of the input operands a, b. Montgomery multiplication and the RSA Tunnel method depend on the length of the modulus n.

The dependence of computation time on length of inputs a, b for the constant length n is shown in Figure 7.1 and Figure 7.2. The schoolbook method and Comba's method are more efficient for small inputs a, b (length ≤ 600 bits) than Montgomery multiplication or the RSA Tunnel (see Figure 7.1). If the length of the modulus remains constant and the length of inputs a, b becomes higher, the situation changes. For 1024 bit inputs the RSA Tunnel is the best option and Montgomery multiplication is the worst (see Figure 7.1). The efficiency of the RSA Tunnel is obvious for bigger modulus (≥ 1024 bits), see Figure 7.2 which confirms the almost constant computation complexity of the RSA tunnel method.

In summary, the RSA Tunnel is optimal for larger inputs (length of $a, b \geq 850$ bits) with the length of modulus $|n| \geq 400$ bits. The schoolbook method is the best choice for the length of inputs $|a|, |b| \leq 300$ bits and Comba's method is appropriate for the length of inputs $300 \leq |a|, |b| \leq 850$ bits. Assuming all public key cryptographic protocols require parameters longer than 1024 bits, the RSA tunnel method is very useful in practice.

More information and results about modular arithmetics and cryptographic operations on restricted devices can be found in the papers [116], [79], [115] and [118].

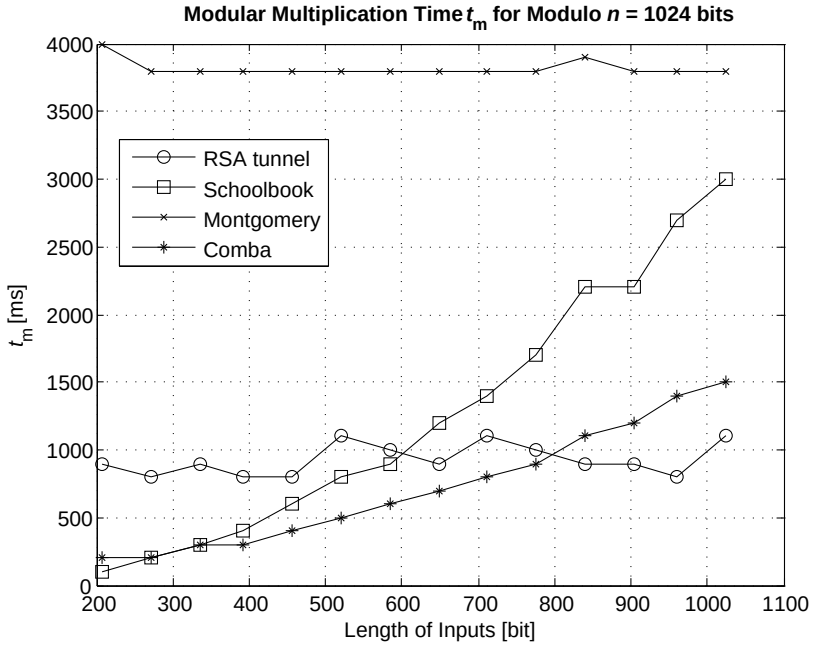


Fig. 6.1: The Modular Multiplication Time for Lower Moduli.

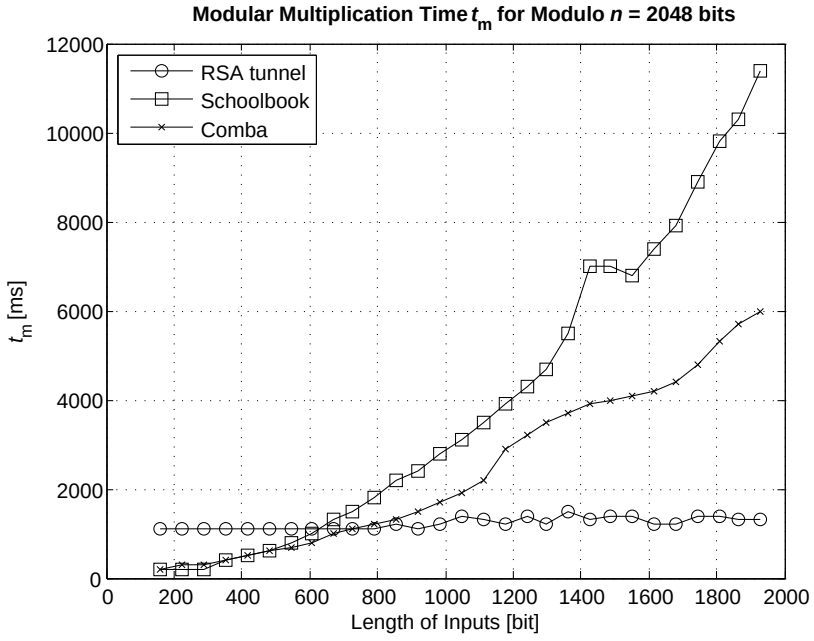


Fig. 6.2: The Modular Multiplication Time for Higher Moduli.

6.2 Optimization Techniques Used in Digital Signatures

The design of cryptographic schemes usually follows two basic goals. Cryptographers try to make schemes as secure as possible, i.e., provably secure under general assumptions. The second goal is communication and computational efficiency. Many cryptographic schemes like digital signatures are adopted in practice to secure information and communication systems. Aimed at digital signatures, these schemes consist of signing and verification phases. Many digital signatures schemes, e.g., RSA [147], ECDSA [93], group signatures (e.g. [20]), etc., provide different properties, such as a signature size, the performance of signing procedure or the performance of verification process. In this section, the optimization techniques of digital signatures such as a signature aggregation and a batch verification are investigated. Aggregate signature schemes save the communication overhead by providing a small size of the chain of signatures. In contrary to this, the batch verification saves the computational overhead by providing the optimized verification process of many signatures. In practice, if the signature size is minimized, the verification process takes usually more operations but if the verification process is optimized, the signature size is usually increased. The results in this section have been presented in the conference paper [121].

6.2.1 Aggregate Signatures

Aggregate signatures can significantly reduce the size of chains of signatures by the aggregation of all signatures into one signature. Due to this fact, the total size of transmitted signatures can be approximately reduced to the size of a single signature. The general principle of aggregate signatures is depicted in Fig. 6.3.

Basic Types of Aggregate Signatures

Generally, there are three kinds of aggregate signatures: multi-signature schemes, non-sequential and sequential aggregate signature schemes. In multi-signature schemes, for example [13], all participants sign the same message.
The definition of **multi-signature schemes**:

Each user U_i signs a same message M to obtain a signature σ_i. A third party can combine a list of signatures σ_i with public keys pk_i into the final single signature $\sum \leftarrow \mathbf{Mulsig}(\sigma_i,\ pk_i, M)$ which is sent to a verifier. Then, the verifier takes the signature $\sum$ and pairs (pk_i, M) to indicate that $\sum$ is valid for i signers with pk_i and the message M, or $\sum$ is invalid.

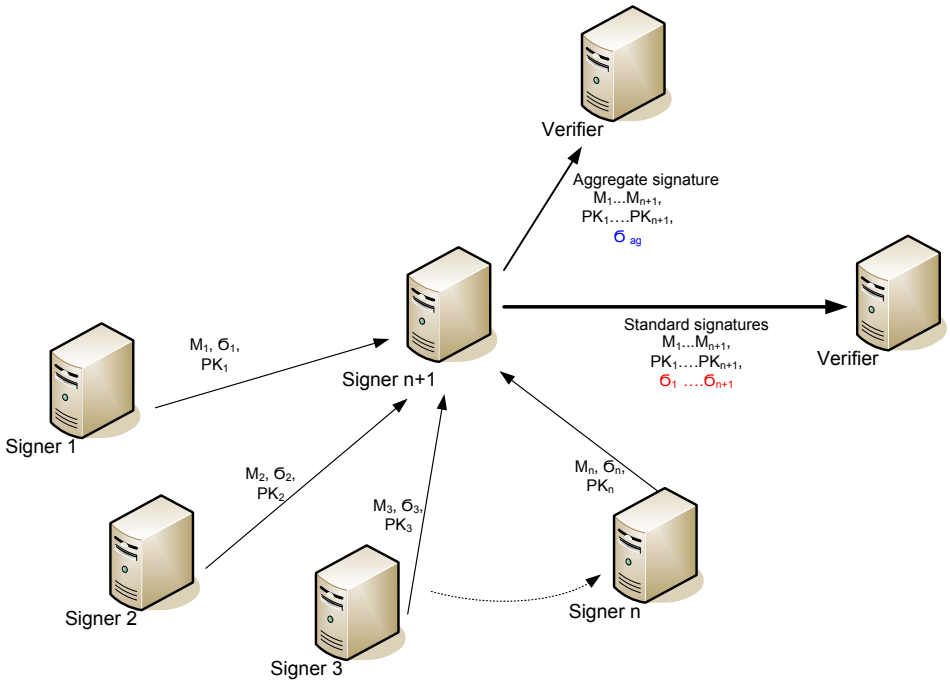


Fig. 6.3: The Basic Principle of Aggregate Signatures.

Aggregate signature schemes generalize multi-signature schemes where several signatures created on distinct messages from different participants are combined into a single signature that has nearly the same size as an ordinary signature. The aggregation can be done by a third party which can be untrusted but he/she has no access to any private key. The aggregate signatures can be non-sequential (defined above) or sequential.

The definition of **non-sequential aggregate signature schemes**:

Each user U_i signs a distinct message M_i with a private key sk_i to obtain a signature σ_i. A third party (an aggregating party/a user) combines a list of signatures σ_i with public keys pk_i into the final single signature $\sum \leftarrow \mathbf{Aggregate}(\sigma_i, pk_i, M_i)$. Then a verifier takes the signature $\sum$ and pairs (pk_i, M_i) to indicate that $\sum$ is valid or not.

The non-sequential aggregate signature schemes aggregate the tuple of signatures, messages and public keys independently on the order of the parties.

The aggregation of sequential schemes is performed sequentially when each signer gets an aggregate-so-far signature and combines this with its own signature.

The definition of **sequential aggregate signature schemes**:

A user U_i checks with the set of public keys **pk** that the current aggregate σ_{i-1}' is a valid signature for the preceding messages **M** where **M** and **pk** are the sequences of messages $\mathbf{M} = (M_1,M_2,...,M_{i-1})$ and public keys $\mathbf{pk} = (pk_1,pk_2,...,pk_{i-1})$. Then, the user U_i signs a distinct message M_i with his/her private key sk_i to obtain a

Tab. 6.4: Comparison of Aggregate Signature Schemes.

Scheme	Security Model	Size of Aggregated Signature	Sign and Aggregate	Verification	Sequential
BGLS [21]	RO	kp	(N+1) mul + 1 exp + 1 hash	(N + 1) pair + N mul + N hash	no
LMRS [112]	RO	kf	(2N + 1) exp + 1 hash	2N exp + N hash	yes
LMRS (with certification) [112]	RO	kf	4N mul + 1 exp + N hash	4N mul + N hash	yes
LOSSW [110]	Standard	2kp	2 pair + (2NL + N + L + 6) mul + 1 exp	2 pair+ NL mul + L mul	yes
Neven [132]	RO	$L, kf + L$	(2N+1) mul + 1 exp + 1 add + 1 hash	2N mul + N add + N hash	yes
Schroder [150]	Standard	4kp	(3N+5) pair + (4N+ 5) exp + (2N+3) mul	(3N+5) pair+ 2N exp + (N − 1) mul	yes
BGR [29]	RO	2kf+256+(128+1)N	1 exp + 2 hash	N exp + 2N hash	no
Yu [175]	RO	2kp	3N mul + N hash + 2(N+1) add	3 pair+N mul + (N-1) add + 2 N hash	yes

aggregate-so-far signature σ_i' and adds it to tuple (σ_i',**M**,**pk**). The last user in the chain makes a final aggregate signature via $\sigma_i \leftarrow$ **Aggregate**(σ_{i-1}', pk_i, M_i). Then, a verifier takes the signature σ_i and the pairs (pk_i, M_i) to indicate that σ_i is valid or not.

Evaluation and Comparison of Aggregate Signature Schemes

The comparison of aggregate signature schemes is listed in Table 6.4. The schemes are usually secured in one from two security models, i.e. the Random Oracle model and the Standard model. To denote the size of aggregate signatures, the parameter kf denotes the size of a RSA group element, e.g. 1024 bits. The parameter kp denotes the size of a elliptic curve element, e.g. 170 bits. The number of participants (signatures) is denoted N. The output length of a collision resistance function is denoted L. It is assumed that the pairing operation (pair) is more computational expensive than exponentiation (exp) and multiplication (mul). The aggregate signing of sequential schemes seems to be more expensive than non-sequential schemes. Nevertheless, the sequential aggregation can be appropriate for many applications such as authenticating the routing information and chaining the certificates.

6.2.2 Batch Verification

Digital signature schemes have a verification phase which is pure check of validity of a signature or a commitment. If a verifier receives N signatures, then he/she can check the validity of these signatures one by one (an individual verification) or he/she can check all N signatures in one instance by a batch verification. In many

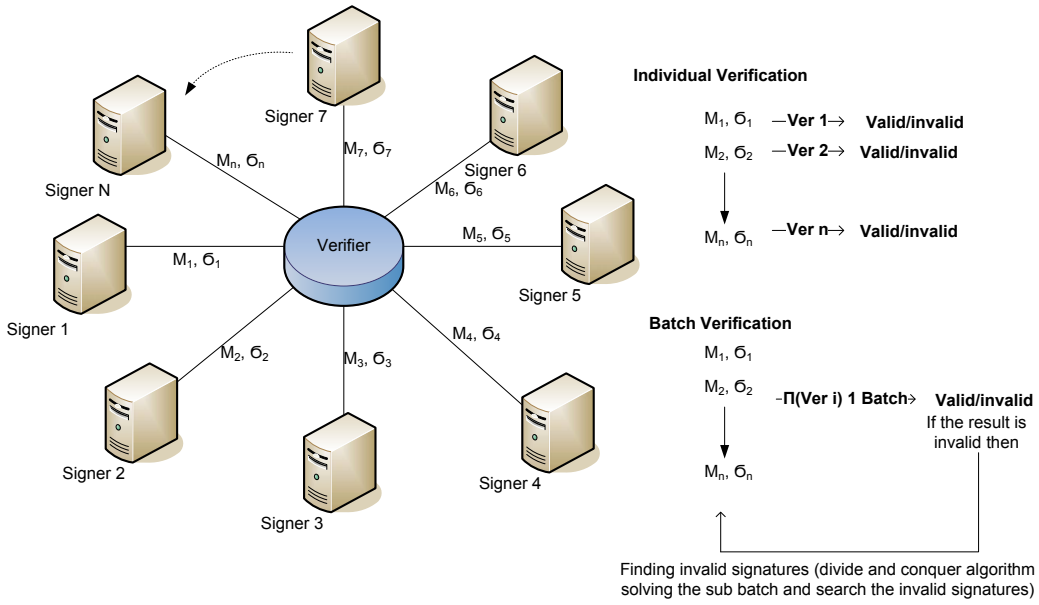


Fig. 6.4: The Basic Principle of Batch Verification.

cases, the batch verification is more efficient than the individual verification. The batch verification is a useful technique especially if verification costs many operations like modular exponentiation, multiplication and/or bilinear pairings. On the other hand, the majority of valid signatures is necessary. That causes better efficiency of the batch verification. For example if fake signatures reach the half of all signatures in a batch, then one instance of batch verification takes more time than all instances of the individual verification.

The definition of the **batch verification of signatures**:

Assuming that the tuple of algorithms (Gen, Sign, Verify) is correct and secure. Then, if Verify(pk_i, m_i, σ_i) = 1 for all $i \in [1, n]$, then BatchVerify((pk_1, m_1, σ_1),...,(pk_n, m_n, σ_n)) = 1. If Verify(pk_i, m_i, σ_i) = 0 for any $i \in [1, n]$, then BatchVerify((pk_1, m_1, σ_1),...,(pk_n, m_n, σ_n)) = 0.

If all N messages in the batch are valid, then the batch verification is valid. If one message or w messages in the batch are invalid, then the batch verification is invalid. In this case, a verifier needs to identify all w invalid messages in the batch. This can cost more operations than the total operations of N individual verifications. Fig. 6.4 depicts the basic principle of the batch verification of digital signatures.

RSA Batch Verification

Fiat [67] presents the batch RSA scheme which is suitable for centralized applications using batch transactions. Nevertheless, if a public exponent used has a reasonable small value, then the individual verification of RSA is quite efficient. Moreover, the

proposed batch verification works only with signatures produced by a single signer.

DSA/ECDSA Batch Verification

Harn [80] presents secure algorithms to verify multiple digital signatures based on the discrete logarithm problem as known as DSA signatures. Instead of verifying each individual DSA signature separately ($2N$ modular exponentiation for N signatures), multiple signatures are verified by the batch verification (2 modular exponentiation for n signatures). Nevertheless, the batch verification works only with one signer, not multiple signers. Moreover, the proposed algorithms cannot identify fake signatures without performing individual verification. Lin et al. [106] deal with the improved batch verification of a DSA variant. They propose batch verification which takes 3 modular exponentiation and $l + n(7 + l/2)$-6 modular multiplications, where l is the bit number of small exponent test. Their scheme does not need any modular inverse. The work [93] deals with the batch verification of original ECDSA signatures. The authors propose several algorithms based upon symbolic manipulations. Nevertheless, their approaches are efficient only for a small batch size, fewer than 7 messages.

Batch Verification of Pairing Based Schemes

The batch verification applied in pairing-based signature schemes reduces the most expensive operations - parings. Ferrara *et al.* [66] summarize the basic batching techniques:

1. Change the verification equation. Check the correct subgroups of elements. Combine all verification equations into one. Employ a small exponentiation test [14].
2. Replace the exponents into the pairing operation, e.g. $e(g_i, h_i)^a \rightarrow e(g_i^a, h_i)$.
3. Combine the pairings with common elements, e.g. $\prod_{i=1}^{n} e(g_i^a, h_i) \rightarrow e(\prod_{i=1}^{n} g_i^a, h_i)$.

The small exponentiation test adds an exponent (e.g 80 bit sized number) to protect against submitting the fake pair elements [14]. The second technique gives a speedup if element g_i is smaller than the element computed by $e(g_i, h_i)$. The third technique reduces N pairings into a constant number. The work [33] presents a batch verification in two short pairing-based signatures so that the total number of pairings is independent on the number of signatures. The proposed approach reduces the total number of pairings by adding the random generation operation and modular exponentiations. The work [114] uses a batch verification to enhance the performance of the group signature scheme BBS04 [20]. Due to this improvement, the verification of N signatures takes only 2 pairing operations instead of $5N$ pairings.

Identification of Fake Signatures in Batch

Fake signatures decrease the efficiency of a batch verification because the whole batch is evaluated as invalid. Fake signatures in the batch can be identified by a divide and conquer algorithm proposed by Pastuszak *et al.* in [136]. The algorithm recursively splits the batch to sub-batches and does a generic test in every round until all fake signatures are identified. This approach reduces batch performance where is $O(1)$ to $O(n\log 2n)$. Ferrara et al. [66] empirically analyze the batch verification of short group signatures based on pairings. The batch verification is applied on the group signature scheme BBS [20]. They results show that if $< 15\%$ of the signatures are invalid then the batch verification with the divide-and-conquer approach is more efficient than an individual verification.

6.2.3 Experimental Results of Optimization Techniques

The implementation of a batch verification and aggregate signature schemes are measured on a machine with Intel(R) Xeon(R) CPU X3440 @ 2.53GHz, 4 GB Ram, Windows 7 Professional. The cryptographic operations are provided by the java.math.BigInteger class and jPBC Library.

The comparison of the signing performance of aggregate signatures is depicted in Fig. 6.5. Besides the aggregate signature scheme proposed by Schroder [150], all schemes take ≤ 1 s for the sign and aggregate of 100 signatures. The most efficient scheme is the scheme BGR [29] using the lazy verification where the sign and aggregate phase is independent on the number of signatures. The comparison of verification performance of aggregate signatures is depicted in Fig. 6.6.

Due to a plenty of pairing operations used in a verification, Schroder's scheme [150] and BGLS scheme [21] take ≤ 1 second if the number of all messages are ≥ 25. The most efficient scheme is the Neven scheme [132] which takes ≤ 8 ms for 100 messages. Nevertheless, the main goal of aggregate signature schemes is to keep a signature size to minimum. The scheme BGLS [21] offers the shortest signature size from schemes compared, it is only 170 bits. The signature aggregation is appropriate in secure routing protocols such as Secure Border Gateway Protocol (SBGP) and in wireless networks with restricted bandwidth.

The performance of the batch verification is compared with the individual verification, see Fig. 6.7. The batch verification significantly optimizes the verification process of N messages in the group signature scheme [20]. The total time of the batch verification takes about 20 % of the total time of the individual verification. Due to this fact, the batch verification is appropriate to apply in Vehicular Ad hoc Networks (VANET), such as in the paper [120], and Many to one Networks (MANET) where one node must verify a lot of messages from many nodes in a short

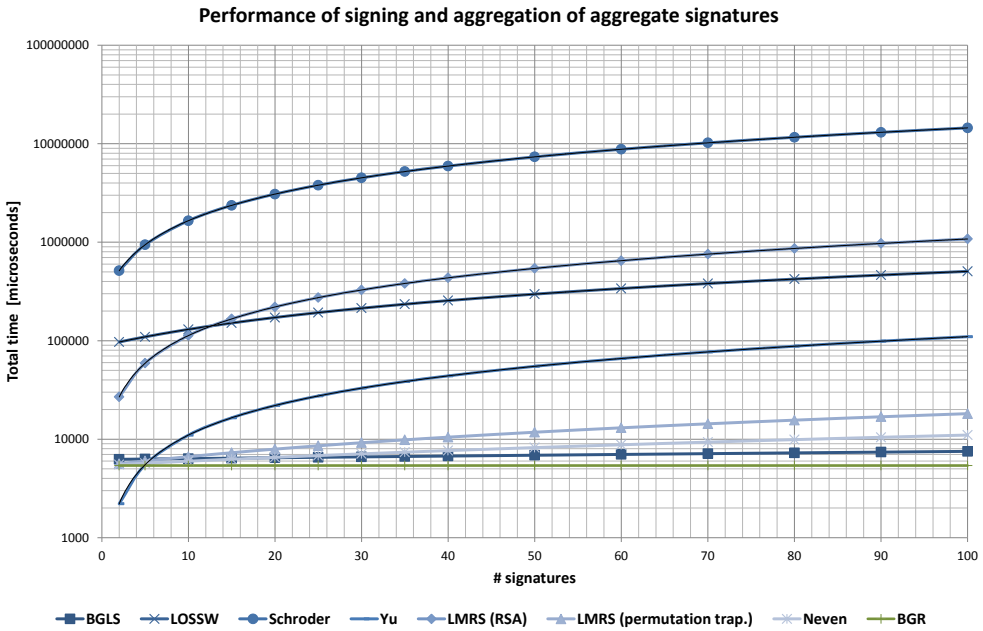


Fig. 6.5: The Comparison of Aggregate Signatures for the Sign and Aggregate Phase.

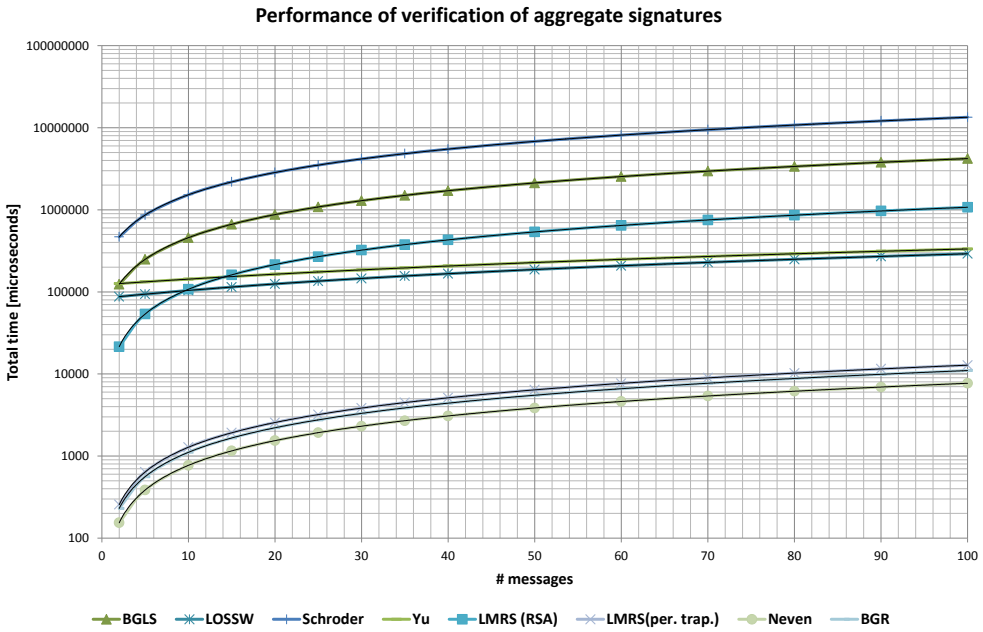


Fig. 6.6: Comparison of Aggregate Signatures for Verification Phase.

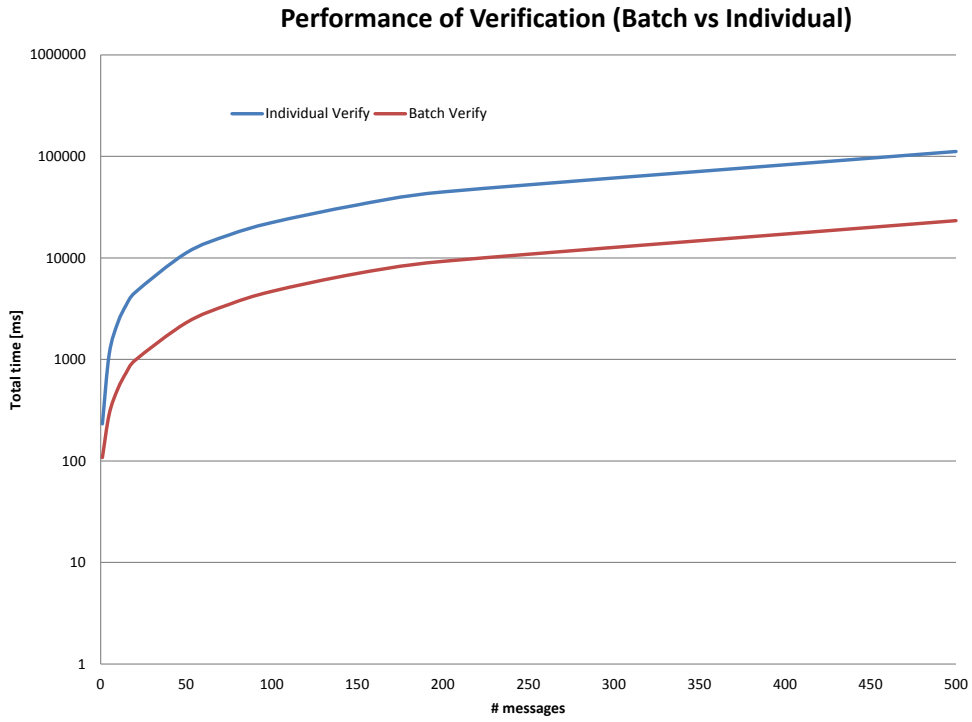


Fig. 6.7: Performance of Batch and Individual Verification Applied on Short Group Signature Scheme.

time period. In general, the batch verification and signature aggregation techniques can be also useful for anonymous authentication and credential schemes such as [31], [77].

6.2.4 Optimization Techniques Applied to Pairing-Based Schemes

This section describes the basic optimization tricks that can be applied to PBC schemes, especially PBC signature schemes.

Pairing Precomputation

If a pairing operation has static values as inputs, then this operation can be computed in advance. Static values are usually generated cryptographic parameters that are not changed during the scheme's lifetime. For example, the signing phase of the BBS04 scheme costs 3 pairing operations without precomputation. On the other hand, pairings $e(h, w), e(h, g_2)$ and $e(g_1, g_2)$ can be precomputed because all inputs to the pairing functions are static. Also, the pairing $e(T_3, g_3)$ can be precomputed, but this approach has no effect in real-time applications, where signatures are computed periodically because T_3 has to be computed for each signature, see Equation 4.1 in Section 4.3.2.

Pairing Collapsing

Pairing collapsing is a technique that enables aggregating pairings into one pairing, see an example in equation 6.3.

$$e(T_3, g_2)^{s_x}(e(T_3, w)^c = e(T_3, g_2^{s_x} w^c) \tag{6.3}$$

Pairing Delegation

The pairing delegation approach can be used if a computational restricted device delegate the computation of pairing $e(A, B)$ to a third party with a more powerful device, e.g. server. The inputs A and B of the pairing operation can be public but there are techniques that enable to compute pairing $e(A, B)$ where A an B are private and by using randomized masking a third party does not learn nothing about these inputs. More about the secure delegation of pairing operations can be found in the paper [51].

6.3 Summary of Chapter 6

The empirical measurement of cryptographic and modular arithmetic operations shows that many of these operations can be computed relatively fast on some of hand-held devices. Nevertheless in case that the devices with CPU frequency of few tens MHz are employed in a communication system, then these devices may have longer computational time of some more expensive operations such as exponentiation and multiplication of large integers. On the other hand, the most demanding operation is the pairing operation which may take several seconds on some computationally constrained devices. To reduce the number of pairings in the protocols, the optimization techniques such as the batch verification, the pairing collapse approach and others can be employed. The optimization techniques applied on digital signature schemes reduce a communication and computation overhead. Aggregate signature schemes add a few cryptographic operations to save a communication overhead. A small communication overhead is demanded for example in routing protocols. The pairing based aggregate signatures have a shorter signature size than RSA based schemes. On the other hand, these pairing-based schemes are more computationally expensive. According to the experimental results, the batch verification of pairing based group signatures optimizes the verification process up to 20 % to the invidual verification. The batch verification techniques significantly decrease the number of pairing operations in a verification. The efficient verification is needed if nodes have to verify several tens or hundreds signatures, e.g. in cloud computing, many to one networks or vehicular ad hoc networks.

7 PROTOCOL 1: PAIRING BASED GROUP SIGNATURE WITH EFFICIENT REVOCATION

This chapter deals with a user revocation in group signatures and presents a novel cryptographic protocol with a group signature scheme with Verifier-Local Revocation (VLR) employing a natural expiration. This work has been presented in the conference paper [119].

The first scheme of a group signature was introduced in [47]. Thereafter, many group signature schemes have been proposed with various parameters and different ways how to revoke group members. Revocation can be divided into three main mechanisms. The first method revokes members by the reinitialization of group public key and sending it to all unrevoked members which must recomputate group member secret keys. This method burdens communication and adds computational operations anytime when a member is added or revoked. The second and more efficient mechanism than the first one is based on sending a single public broadcast message to all members without need to recompute secret keys. This accumulator-based revocation method is mostly used as a white-list revocation but can be used as a black-list revocation as well. Users must prove their validity proofs called witnesses and that are included on a white-list accumulator (or not present on a black-list accumulator). Verifiers do not need any revocation list. Nevertheless, signers have to keep track of the changes of the accumulator and have to be online. This approach is more convenient for verifiers than for signers. The third option how to check the revoked users is to employ a list with revoked users (keys, tokens credentials etc.) maintained by GM. GM sends it to verifiers who must perform a revocation check. This method is called Verifier-Local Revocation (VLR). Group members do not have to track any updates or refresh their witnesses. Since the members have no work with revocation check, this check must be computed by the verifier. VLR solutions provide less interactivity so signer can be off-line and has no additional computation compared to accumulator-based solutions. The drawback of VLR solutions is usually the growth of revocation lists to enormous sizes in a large group. Hence, the revocation check is too expensive for verifiers and the reinitialization of parameters and keys have to be done.

Revocation based on accumulators, presented in [34], and verifier-local revocation, such as in [23], are two revocation techniques having different pros and cons. For example the work [65] proposes a group signature scheme based on [20] using the pairing-based dynamic accumulator introduced in [34]. Every group member has to refresh his/her witness and an accumulator, otherwise a verifier rejects his/her sig-

nature. Users have to be online and download refreshed accumulators from a group manager. The solution works only if users and verifiers are online and has the same distribution of accumulators.

The protocol proposed in this thesis is aimed to immediate revocation which is also suitable for off-line signers in non-large groups. The solution is based on the verifier-local revocation approach. A group signature scheme with VLR employing a natural expiration is proposed to reduce the length of revocation list by time. The proposal focuses on efficiency in the signing phase and the verification phase including the revocation check.

7.1 Revocation in Group Signatures

The verifier-local revocation introduced in [23] can be an efficient revocation solution for signers. The signatures after revocation become linkable which can be inadvisable for some applications. Therefore, the works, e.g. [131], [105] or [28], add a property called Backward Unlinkability (BU). In the paper [131], the authors extend a group signature scheme [23] and add BU. They employ the revocation tokens of revoked members for certain time intervals to ensure that former signatures cannot be linkable if the member is revoked. While the proposal [131] is proven in the random oracle model, the work [105] presents the VLR group signature scheme with BU that is proven in the standard model. Nevertheless, the revocation check also costs 1 pairing operation per one revocation token as in [131]. To improve computational overhead, one revocation check is reduced from one pairing to one exponentiation in the paper [48]. In the paper [28], the scheme proposed in [48] is patched to satisfy backward unlinkability, traceability and exculpability in the random oracle model. Time intervals that are used in [131], [105], [48] and [28] can moderate the size of RL. If the time interval is too long then the revocation list is too large, otherwise, if the time interval is too short then the group public key and group member secret keys are too long.

CKS 2010 [35] present revocation with efficient updates. The validity time of a credentials is encoded into an attribute. Nevertheless, the solution does not support an immediate revocation. In time-critic services, the solution has to be combined with an accumulator solution. Due to the time validity of a credential, the accumulator keeps limited size and the number of an accumulator and witness updates is lower than no-time-restricted credentials. Anyway, users have to be online to keep the newest version of accumulator. The authors claim that this approach cannot be used on group signatures since a validity time period identifier is hard to include. However, they suggest to sign the second message as an epoch identifier.

The work [54] proposes a pairing-based group signature scheme with VLR employing time-bound secret keys and without BU. Each group secret key has an expiration date so the verifier checks the revocation list that excludes expired members. Only one exponentiation is needed to check whether the key is revoked. Nevertheless, the scheme performs seven pairing operations per one message in the verification phase.

As in [54] also the protocol proposed in this chapter does not provide BU, thus, it can be inappropriate for applications that demands backward linkability. The proposed protocol which is based on BS scheme [23] provides more efficient verification than the scheme in [54] and related VLR group signature schemes due to the batch verification. Moreover, to ensure the shorter revocation tokens, the proposed solution uses time offsets compared to using date formats in [54].

7.2 Preliminaries of Protocol 1

In this section, the cryptography background and system model are outlined.

7.2.1 Cryptography Used

The proposed scheme is based on a group signature scheme proposed by Boneh and Shacham (the BS04 scheme) [23] with verifier-local revocation that ensures anonymity, authenticity, message integrity, non-repudiation, unlinkability and traceability. The scheme uses bilinear maps and is based on the q-SDH problem and Decision Linear problem, which have been described in [23]. This scheme is modified to ensure more efficient verification algorithm by a verifier-local revocation with time-bound group member secret keys and batch verification. To make time-bound group secret member keys, the methods called 0-encoding/1-encoding presented in [54] are employed.

Bilinear Pairings

The scheme is based on bilinear pairing operations like in [23]. The notification used is as follows:

- G_1 is a multiplicative cyclic group of prime order p.
- G_2 is multiplicative group of exponent p.
- G_T is multiplicative cyclic group of order p.
- g_1 is a generator of G_1 and g_2 is a generator of G_2.
- ψ is computable homomorphism from G_2 to G_2, with $\psi(g_2) = g_1$.
- e is a computable bilinear map $e : G_1 \times G_2 \rightarrow G_T$ with the following properties:

– Bilinearity: for all $u \in G_1, v \in G_2$ and $a, b \in \mathbb{Z}, e(u^a, v^b) = e(u, v)^a b$.
– Non-degeneracy: for all $e(g_1, g_2) \neq 1$.

0/1-ENCoding

The 0-encoding and 1-encoding reduce the *greater than* predicate to *set intersection* predicate by converting a date format in binary string to a value in Z_p. To convert elements of binary strings to a value in Z_p, the procedure presented in [54] is used. The procedure is defined as follows:

1. Use the 0/1-ENCoding of a l-bit binary string $t = t_{[l]}t_{[l-1]}...t_{[1]}$, where t is a date encoded in binary string and $t_{[i]}$ denotes i-th bit of t, by
   $T_t^0 = \{t_{[l]}t_{[l-1]}...t_{[i+1}1 \| t_{[i]} = 0, 1 \leq i \leq l\}$,
   $T_t^1 = \{t_{[l]}t_{[l-1]}...t_{[i} \| t_{[i]} = 0, 1 \leq i \leq l\}$.

   Based on the theorem in [107], $x > y$ iff T_x^1 and T_y^0 have a common element.

2. Ensure that the sets start with '1' by adding '1'
   $\overline{T_t^0} = \{1 \cdot 10^{l-i+1} + t_{[l]} \cdot 10^{l-i} + t_{[l-1]} \cdot 10^{l-i-1}...t_{[i+1} \cdot 10^1 + 1 \| t_{[i]} = 0, 1 \leq i \leq l\}$,
   $\overline{T_t^1} = \{1 \cdot 10^{l-i+1} + t_{[l]} \cdot 10^{l-i} + t_{[l-1]} \cdot 10^{l-i-1}...t_{[i+1} \cdot 10^1 + t_{[i]} \| t_{[i]} = 1, 1 \leq i \leq l\}$.

3. Fill up the sets with incomparable dummy elements to achieve an equal number of elements: $\{t_l, t_{l-1}, ..., t_1\} \leftarrow$ 0-ENC(t), where $t_i \leftarrow \{z$ if $z \in \overline{T_t^0} \wedge 2 \cdot 10^i$ otherwise $\}$ and $\{t_l, t_{l-1}, ..., t_1\} \leftarrow$ 1-ENC(t), where $t_i \leftarrow \{z$ if $z \in \overline{T_t^1} \wedge 3 \cdot 10^i$ otherwise $\}$.

Here, an example assuming two dates y='1301' and x='1303' (2013-January and 2013-March) in a date format 'YYMM' is outlined. It can be shown that the 0/1-ENCoding indicates which of date is the newer one and if common element appears then $x > y$. The date '1301' and '1303' are encoded into binary strings as y=101 0001 0101 and x=101 0001 0111. Nevertheless, a time offset based on number of months from present can map much longer time period for the same length of bits.

The 0/1-ENCoding is employed on $x = 10100010111, y = 1010001010$ and output is:

$T_y^0 = \{11, 1011, 10101, 101001, 10100011, 1010001011\}$,
$T_x^1 = \{1, 101, 1010001, 101000101, 1010001011, 10100010111\}$,

$\overline{T_y^0} = \{111, 11011, 110101, 1101001, 110100011, 11010001011\}$,
$\overline{T_x^1} = \{11, 1101, 11010001, 1101000101, 11010001011, 110100010111\}$,

0-ENC(y) $\rightarrow$ {20, 111, 2000, 11011, 110101, 1101001, 20000000, 110100011, 2000000000, **11010001011**,200000000000},
1-ENC(x) $\rightarrow$ {11, 300, 1101, 30000, 300000, 3000000, 11010001, 300000000, 1101000101, **11010001011**,110100010111}.

It is clear if the element **11010001011** is common for both sets then $x > y$ is true. The sketch of proof can be found in [54].

7.2.2 System Model of Protocol 1

The designed system model consists of three parties:

- Group manager (GM). It is assumed that GM is a trusted party. GM initializes all group signature parameters, one group public key, one group manager secret key and group member secret keys. GM also manages a revocation list which includes revoked users.
- Verifier (V). V checks only signed messages by a group public key and if user is on the revocation list or not.
- User (U). U, who correctly joins into a group, can sign any message by his/her group member secret key and send it to V.

7.3 Description of Protocol 1

In this section, the proposed protocol is outlined. The protocol consists of five main phases: setup, join, sign, verify and open. The protocol is based on BS04 group signature scheme [23] and it is enhanced on the efficient group signature scheme with time-bound secret keys with batch verification.

7.3.1 Setup Phase of Protocol 1

In the `setup algorithm` **Setup**(λ) $\rightarrow$ ($parameters$, $gpk, gmsk$), GM sets group signature parameters, group public key and group manager secret key as follows:

- Based on the length of the security parameter λ, the group signature parameters $G_1, G_2, g_1, g_2, \psi, e$ are established since $g_1 = \psi(g_2)$ if $e(\psi(g_2), g_1) \neq 1$.
- GM generates the group manager secret key $gmsk = (\gamma)$ where $\gamma \overset{R}{\leftarrow} Z_p$.
- The group public key $gpk = (g_1, g_2, w)$ is published where $w = g_2^{\gamma}$.

7.3.2 Join Phase of Protocol 1

In the join algorithm, the i-th user U_i joins into a group which is managed by a group manager GM. The `join algorithm` **Join**($RL, gpk, gmsk$) $\rightarrow$ gsk_{U_i} is performed

between GM and U_i as follows:

- Based on the variable values such as the length of revocation list, the reputation of U_i etc., the group manager decides about the duration of expiration date τ_i for the group member secret key gsk_{U_i}.
- GM encodes the expiration date τ_i by the 1-Encoding: $\{\tau_{ij}\}_{j\in[1,l]} \leftarrow$1-Enc$(\tau_i)$ where l is the length of date format.
- For $(j = 0; j \leq l; j++)$, GM computes $A_{ij} = g_1^{\frac{1}{\tau_{ij}x_i+\gamma}}$, where $x_{ij} \overset{R}{\leftarrow} Z_p^*$ and $\tau_{ij}x_{ij} + \gamma \neq 0$
- GM sends user's group member secret key $\tau_i, \{A_{ij}, x_{ij}\}$, the group public key and public parameters via secured connection to user (e.g. via TLS). The revocation token $\tau_i, \{x_{ij}\}$ is saved.
- U_i encodes the expiration date τ_i by the 1-Encoding: $\{\tau_{ij}\}_{j\in[1,l]} \leftarrow$1-Enc$(\tau_i)$ and checks $e(A_{ij}, w^{\tau_{ij}} g_2^{x_{ij}}) = e(g_1, g_2)$ for each $j \in \{1, 2, ..., l\}$ if gsk_{U_i} is valid.

7.3.3 Signing Phase of Protocol 1

Every user U_i who wants to send a new message to a verifier has to sign the message. Every U_i has a member secret key $gsk_{U_i} = \tau_i, \{A_{ij}, x_{ij}\}$ and a group public key $gpk = (g_1, g_2, w)$. U_i signs a message $M \in (0,1)^*$ and outputs the signature of knowledge $\sigma = (t_{cur}, k, T_1, T_2, c, s_\alpha, s_x, s_\delta, R_2)$.

The `Signing algorithm` **Sign**$(M, gsk_{U_i}, gpk, t_{cur}) \rightarrow \sigma$ is performed by U_i as follows:

1. U_i checks if his/her gsk_{U_i} is not expired by $t_{cur} < \tau_i$, where t_{cur} is a current date (e.g. a current month or a current date in format 'YYMMDD' as in [54]) or the date of the signature expiration. If $t_{cur} \geq \tau_i$, the algorithm halts.
2. The dates are converted into *intersection check* by the 0/1-Encoding: $\{\tau_{ij}\}_{j\in[1,l]}$ $\leftarrow$1-Enc(τ_i) and $\{t_j\}_{j\in[1,l]} \leftarrow$0-Enc$(t_{cur})$ where l is the length of date format used.
3. The index $k \in \{1, 2, ..., l\}$ is found such that $\tau_{ik} = t_k$ and the pair of A_{ik}, x_{ik} from gsk_{U_i} is selected.
4. U_i chooses random elements $\alpha, r_\alpha, r_x, r_\delta \in Z_p^*$.
5. U_i computes the group signature by the following steps:
   Firstly, U_i sets
   $$(\overline{u}, \overline{v}) = H_0(M, gpk, t_{cur}), \tag{7.1}$$
   where H_0 is two-dimensional hash function, mapping $\{0,1\}^*$ to G_2^2. Then, the user computes the images in G_1 by
   $$(u, v) = \psi(\overline{u}, \overline{v}) \tag{7.2}$$

pseudonyms by

$$T_1 \;=\; u^{x_{ik}}, T_2 \;=\; A_{ik} v^{\alpha}, \tag{7.3}$$

helper values by

$$\delta \;=\; \alpha x_{ik}, \tag{7.4}$$

$$\begin{aligned} R_1 &= u^{r_x}, \\ R_2 = \; e(T_2, g_2)^{-r_x} e(v, g_2)^{r_\delta} \; e(v, w)^{r_\alpha r_{ik}} &= \\ e(T_2^{-r_x} v^{r_\delta}, g_2) \; e(v, w)^{r_\alpha r_{ik}}, & \\ R_3 &= T_1^{r_\alpha} u^{-r_\delta}, \end{aligned} \tag{7.5}$$

a challenge value by

$$c = H(gpk, t_{cur}, M, T_1, T_2, R_1, R_2, R_3), \tag{7.6}$$

and response values by

$$\begin{aligned} s_\alpha &= r_\alpha + c\alpha, \\ s_x &= r_x + c x_{ik}, \\ s_\delta &= r_\delta + c\delta. \end{aligned} \tag{7.7}$$

6. U$_i$ sends the message M with the signature $\sigma = (t_{cur}, k, T_1, T_2, c, s_\alpha, s_x, s_\delta, R_2)$.

7.3.4 Verification Phase of Protocol 1

The verifier (V) verifies messages received from pseudonymous users. V checks the group signature, the time validity of the signature and if a pseudonymous user who signed the received message is not in a Revocation List (RL).

Individual Verification

The `Individual verification algorithm` **InVerify** $(M, gpk, \sigma, t_{act}, RL) \rightarrow$ valid/invalid is performed by V as follows:

1. The time validity of the signature is checked by $t_{act} > t_{cur}$, if *yes* then the algorithm halts. To continue the algorithm, the value t_{cur} must be equal or newer than actual date t_{act} measured by verifier.
2. The date t_{cur} is converted into the *intersection check* by the 0-Encoding: $\{t_j\}_{j\in[1,l]} \leftarrow$0-Enc$(t_{cur})$ and by k from the signature is found t_k.
3. V restores u, v:

$$(\overline{u}, \overline{v}) = H_0(M, gpk, t_{cur}), \tag{7.8}$$

where H_0 is a two-dimensional hash function, mapping $\{0,1\}^*$ to G_2^2. Then, the user computes the images in G_1 by

$$(u, v) = \psi(\overline{u}, \overline{v}) \tag{7.9}$$

4. V restores $\overline{R}_1$ and $\overline{R}_3$:

$$\overline{R}_1 = u^{s_x} T_1^{-c}, \overline{R}_3 = u^{-s_\delta} T_1^{s_\alpha}. \tag{7.10}$$

5. V computes a new control hash c' from the received parameters:
   $c' = H(gpk, t_k, M, T_1, T_2, \overline{R}_1, R_2, \overline{R}_3)$.
   and checks if $c' = c$. If yes, then V continues with the verification, otherwise the message is inconsistent and is refused.
6. V checks if

$$\begin{aligned} R_2 &= e(T_2, g_2)^{-s_x} e(v, w)^{(t_k s_\alpha)} \\ e(v, g_2)^{(s_\delta)} &(e(g_1, g_2) e(T_2, w^{t_k})^{-1})^c \\ &= e(T_2^{-s_x} v^{s_\delta} g_1^c, g_2) e(v^{s_\alpha} T_2^{-c}, w^{t_k}) \end{aligned} \tag{7.11}$$

7. The signed message is valid if Equations 7.11 hold.

8. The verification phase continues by a revocation check in the following subsection.

Revocation Check

The verifier opens the actual revocation list $RL = (\tau_i, \{x_{ij}\})$ containing r revoked tokens where $j \in [1, l]$ (l is the length of the date format used) and $i \in [1, r]$ to check if the signed message is received from a revoked or an unrevoked user. The `Revocation check algorithm` **RevCheck**$(RL, \sigma) \rightarrow$ revoked/unrevoked is performed as follows:

- For each i-pair of $\tau_i, \{x_{ij}\}$, V recomputes by the 1-Encoding: $\{\tau_{ij}\} \leftarrow$1-Enc(τ_i) and find an index m $(1 \leq m \leq l)$ such that $\tau_{im} = t_k$, selects x_{im} from RL and checks if

$$T_1 = u^{x_{im}}. \tag{7.12}$$

- If Equation 7.12 holds then user's signed message will be discarded because the i-th user with x_{im} has been revoked by GM.

If a new user is revoked then GM sends to verifiers the refreshed revocation list. Further, every verifier discards old records with obsolete pairs $\tau_i, \{x_{ij}\}$ to reduce the length of RL.

Batch Verification

If V receives more messages in one short period then V verifies these signed messages in one batch.

The `Batch Verification algorithm` **BatchVerify**$(M_1, M_2, .., M_n, \sigma_1, \sigma_2, .., \sigma_n, gpk, t_{act}, RL) \rightarrow$ valid/invalid.

V uses $gpk = (g_1, g_2, w)$ to verify n messages with $\sigma_z = (t_{zcur}, k_z, T_{z1}, T_{z2}, R_{z2}, c_z, s_{z\alpha}, s_{zx}, s_{z\delta})$ for $z = 1, ..., n$, does:

1. V checks the time validity (of a signature) by $t_{act} > t_{zcur}$, if *yes* then the algorithm aborts. To continue the algorithm, the value t_{zcur} must be equal or newer than actual date t_{act} measured by the verifier.
2. The date t_{zcur} is converted into *intersection check* by the 0/1-Encoding: $\{t_{zj}\}_{j\in[1,l]}$ $\leftarrow$0-Enc(t_{zcur}) and by k_z from the signature is found t_{zk}.
3. V restores u_z, v_z:

$$(\overline{u_z}, \overline{v_z}) = H_0(M_z, gpk, t_{zcur}) \tag{7.13}$$

   where H_0 is a two-dimensional hash function, mapping $\{0,1\}^*$ to G_2^2. Then, the user computes the images in G_1 by

$$(u_z, v_z) = \psi(\overline{u_z}, \overline{v_z}) \tag{7.14}$$

4. V restores $\overline{R}_{z1}$ and $\overline{R}_{z3}$:

$$\overline{R}_{z1} = u_z^{s_{zx}} T_{z1}^{-c_z}, \overline{R}_{z3} = u_z^{-s_{z\delta}} T_{z1}^{s_{z\alpha}}, \tag{7.15}$$

5. V computes a new control hash c'_z from the received parameters:
   $c'_z = H(M_z, gpk, t_{zcur}, T_{z1}, T_{z2}, \overline{R}_{z1}, R_{z2}, \overline{R}_{z3})$,
   and checks if $c'_z = c_z$. If yes, then V continues with the verification, otherwise the message with the signature is inconsistent and is refused.
6. V randomly selects $\theta_1, \theta_2, ..., \theta_n \in Z_p$ with l_b bit, checks the batch if

$$\begin{aligned} \prod_{z=1}^{z=n} R_{z2}^{\theta_z} = e(\prod_{z=1}^{z=n} (T_{z2}^{-s_{zx}} v_z^{s_{z\delta}} g_1^{c_z})^{\theta_z}, g_2) \\ e(\prod_{z=1}^{z=n} (T_{z2}^{c_z} v_z^{-s_{j\alpha}})^{\theta_j}, \prod_{z=1}^{z=n} (w^{t_{zk}})) \end{aligned} \tag{7.16}$$

7. The batch with signed messages is valid if Equations 7.16 hold.
8. V performs `Revocation check algorithm` to ensure that there are no messages from already revoked users.

It can be noted from Equations 7.11 and 7.16 that the individual verification costs 2 pairing operations per one message but the batch verification costs only 2 pairing operations per n messages.

In case the batch verification is valid, then all messages from the batch are valid. In case the batch verification fails, then the *divide-and-conquer* approach is used to identify the invalid signatures that can be discarded. At the end of the *divide-and-conquer* approach, the final two messages are individually verified.

7.3.5 Open Phase of Protocol 1

GM stores revocation tokens $\tau_i, \{x_{ij}\}$ of all users. Every correctly signed message M with the group signature σ and the group public key can be opened by GM. User index i which is connected with a user ID stored in a database can be revealed by `Revocation check algorithm`. If the revealed user has still the unexpired group member secret key then GM puts this user onto the revocation list and send refreshed RL to verifiers.

7.4 Evaluation and Results of Protocol 1

This section evaluates the proposed protocol and compares it with the related work. Further, an experimental implementation and initial results of the scheme are outlined. The proposed scheme is based on the BS04 scheme [23] and inherits all security assumptions of [23].

GS scheme:	Designed scheme	BS04 [23]	CLHZ12 [54]	NF07 [131]	BP11 [28]
Batch:	yes	no	no		
Length of signature:	$2G_1, G_T, 4Z_p$ (2059 bits)	$2G_1, 5Z_p$ (1192 bits)	$4G_1, 5Z_p$ (1549bits)	$3G_1, 6Z_p$ (1533 bits)	$5G_1, \lambda+6Z_p$ (23301 bits)
Verification of n messages with r revoked users in RL:					
Pairings	**2**	$3n + 2nr$	$7n$	$2n + 2nr$	$1n$
Exponentiation	$10n+ 1nr$	$6n$	$13n + 1nr$	$6n$	$3n\lambda+1nr+5n$
Multiplication	$9n+1$	$6n+1nr$	$9n$	$6n+1nr$	$2n\lambda+8n$
Signing:					
Pairings	2	2	5	1	1
Exponentiation	8	8	12	7	16
Multiplication	9	9	10	8	$10+\lambda$

Tab. 7.1: Performance Evaluation of VLR Group Signature Schemes - Signing and Verification Phases.

7.4.1 Evaluation and Comparison

The protocol is evaluated in the main phases: signing and verification which includes revocation. Table 7.1 depicts the comparison of the protocol with related solutions BS04 [23], CLHZ12 [54], NF07 [131] and BP11 [28]. To be noted that the verification of n messages also includes the revocation check of r revoked users. Assuming that p

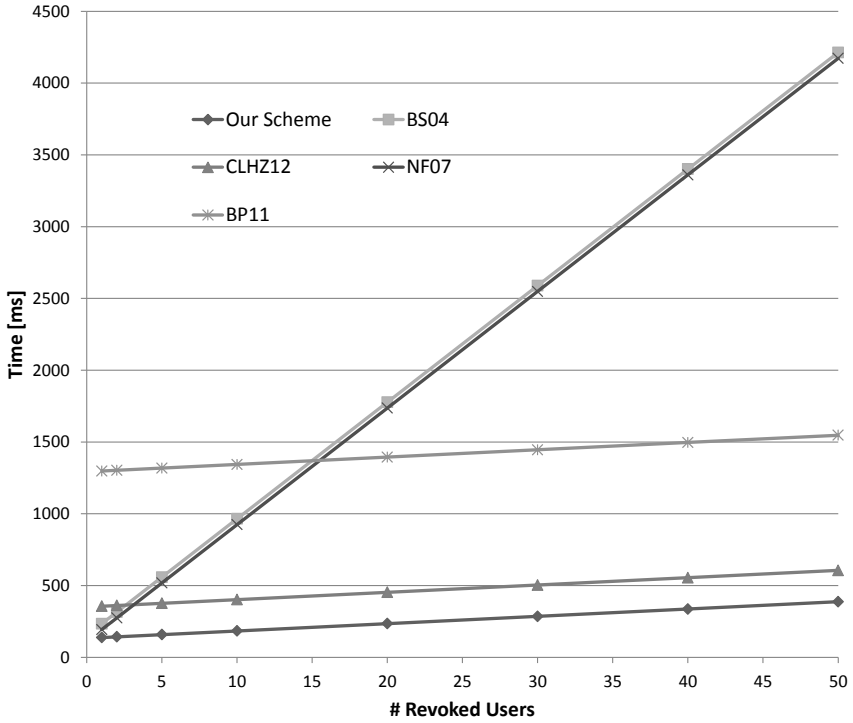


Fig. 7.1: Performance of Verification for 1 Signature.

is a 170-bit prime, the length of elements in G_1 is 171 bits and the length of elements in G_T is 1020 bits. The protocol uses the date format for 255 months (21 years) formed in an offset since the setup of system. Then, the date format and index k take only 11 bits (8 bits for date, 3 bits for index k). The designed scheme produces 2059-bit signatures. Comparing with the revocation token used in CLHZ12 [54] which has 14 elements, the revocation token has only 8 elements in the protocol. In BP11 scheme [28], the size of λ is 80 which afflicts the length of a signature (23301 bits). Due to the batch verification applied in the protocol, the verification takes only 2 pairings per n messages.

7.4.2 Experimental Results

To obtain initial results, the proposal have been implemented as a proof of concept application in JAVA. The main core of the experimental implementation is formed by the group signature scheme that uses the Java Pairing Based Cryptography (jPBC) Library[1]. The implementation employs the MNT curves type D with the embedding degree $k = 6$, the 171-bit order of curves and the pre-generated parameters d840347-175-161.param. The implementation is tested on a machine with Intel(R) Xeon(R) CPU X3440 @ 2.53GHz, 4 GB Ram, Windows 7 Professional. In the scheme, the

[1](available on http://gas.dia.unisa.it/projects/jpbc/index.html)

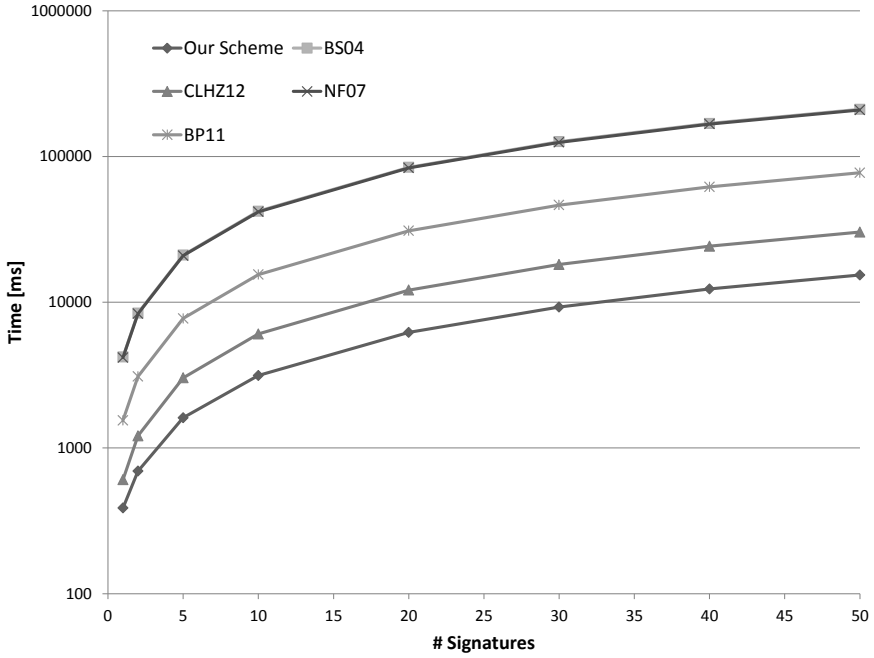


Fig. 7.2: Performance of Verification with 50 Revoked Users.

signing phase of one message takes approx. 120 ms and one verification with empty RL takes 132 ms. The revocation check with one revocation token in the list takes 5,1 ms. In Figures 7.1 and 7.2, the performance of the verification phase of the scheme and related schemes is depicted. The Figure 7.1 shows the performance of verification of 1 signature with growing the number of revoked users. The Figure 7.2 depicts the performance of verification with the size of RL $|\mathrm{RL}| = 50$ with growing the number of signatures. Figures 7.1 and 7.2 confirm that the verification phase in the protocol is more efficient than the verification phase in the related schemes for a variable number of messages and revoked users. The proposed protocol is approximately twice more efficient than the CLHZ12 scheme [54].

7.5 Summary of Chapter 7

The chapter presents the GS scheme with VLR using a natural expiration that can be useful for many applications without back unlikability. The proposed protocol can be applied in services used by the middle-sized groups of users who are off-line. The protocol uses the batch verification to enhance the performance in the verification phase. Hence, verifiers are able to check more signatures at once and save their computational overhead. According to the experimental results, the protocol is more efficient than the related schemes in the verification for the various number of signed messages and revocation tokens placed in the revocation list.

8 PROTOCOL 2: PAIRING BASED GROUP SIGNATURE WITH CATEGORIZED BATCH VERIFICATION

To fulfill the goal of this thesis, this chapter presents a cryptographic protocol that ensures privacy and security in ad hoc networks where the heterogeneous devices such as smartphones, embedded devices and servers are used. Protocol 2 is based on pairing-based group signatures and a categorized batch verification. The protocol ensures the privacy protection of users during data communication and protects data integrity and authenticity. The first version of the protocol has been presented in the conference paper [114]. The extended version has been accepted to publish as the journal paper (IF=1.027) in [123]. The protocol is based on the group signature scheme BBS04 [20]. The proposed solution provides the efficient signing phase and the verification phase. The group signature scheme is designed to keep a long-term unlinkability to ensure user privacy and a short-term linkability to increase the performance of the scheme. Due to a batch verification and the short-term linkability, it is possible to verify many signed messages in one batch. The batch verification reduces the number of bilinear pairing operations e from $n*k$ to l, where n is the number of messages, k is the number of bilinear pairing operations during an individual message verification and l is the number of bilinear pairing operations during the batch verification. In practice, the parameter l is usually lower than k. The batch verification can be computed by equations (8.1) and (8.2), where f_i, h_i, c_i are parameters (points on a elliptic curve) for each i message from the total number of messages n, and, A is a constant (eg. $A = 1$).

$$e(\prod_{i=1}^{n} {f_i}^{c_i}, h_i) = A \tag{8.1}$$

$$\prod_{i=1}^{n} e({f_i}^{c_i}, h_i) = a \ \rightarrow \prod_{i=1}^{n} e(f_i, h_i)^{c_i} = A \tag{8.2}$$

On the other hand, if a message in the batch is invalid, the computational complexity of the batch verification, which is linear in case of presence n valid messages, degrades to logarithmic. If a message is invalid, then the batch verification is also invalid. The batch is split to two batches that are verified again separately. This procedure is performed until all invalid messages are detected.

Security solutions must be as efficient as possible, especially, in real time VANET applications. In urban areas, vehicular ad hoc networks work with large number of users and messages. The individual verification of group signature can take few tens of milliseconds on an embedded device which serves as an on-board computer in

vehicles. If tens messages are verified at one time, the verification phase must be very efficient and fast. Only the batch verification can provide this efficiency but the messages must be valid in most cases.

The protocol in this chapter proposes efficient signing and verification phases to satisfy demanded efficiency by real time VANET applications or large VANETs. Due to the proposed short-term linkability property, it is able to reduce 3 bilinear pairing operations, which is a common number in many related works and the scheme [20], to 0 bilinear pairing operations, 10 exponentiation to 9 and 14 multiplication to 9 in the signing phase. Moreover, the solution proposes a categorized batch verification that is able to detect potential valid messages due to short-term linkability. Thus, these messages are processed with a higher priority. The categorized verification can also resist to some denial of service attacks and the Sybil attack [60].

8.1 Vehicular Ad hoc Network Security

Vehicular network security plays a key role in situations such as the generation of bogus and/or malicious messages, misusing at roads, eavesdropping etc. Common solutions, e.g., [83], [145] guarantee the message integrity, authentication and non-repudiation. Furthermore, privacy is required due to the possibility of drivers being tracked by malicious observers. VANETs can serve in a urban traffic where hundreds of vehicles communicate following the V2V or V2I paradigms, so that the security overhead and computation time are minimal. There is a lot of solutions in VANETs that are secure and keep users' privacy. Nevertheless, privacy-preserving solutions can be vulnerable against several denial of service attacks. The following scenario demonstrates the current security problems which affect the solutions that provide user privacy in VANETs.

Scenario 1: A driver, Alice (A), with the car no. 2, which is depicted in Fig.8.1, records special events (accidents, traffic jams, roads under construction etc.). Depending on the type of event, A immediately broadcasts a warning message through the wireless V2V communication to all the cars which form the VANET. In this scenario, an accident is depicted in Fig.8.1. Let us assume that another driver, Bob (B), with car no. n-1, who is in range and coming closer to A, receives this message. B also receives more messages from other cars in the area. Moreover, other messages can contain contradictory warnings or malicious/bogus information. In a short time, B must consider the validity of these messages and quickly decide changing the route (from planed I. to II.). If B makes the right decision, he can avoid the situation referenced by the first warning message. It is obvious that the decision must come in real time and as soon as possible. Nevertheless, the received

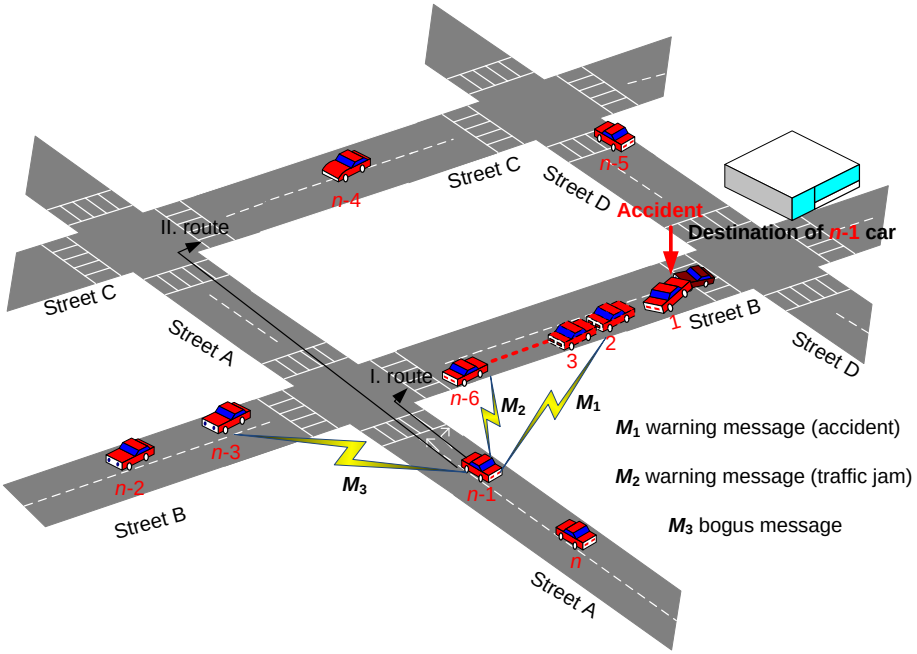


Fig. 8.1: VANETs in Urban Traffic - *Scenario 1.*

messages are from anonymous nodes so B may wonder which messages are coming from honest sources and which are not. The protocol designed in this chapter is based on the employment of a group signature scheme which adds new properties, namely, the short-term linkability and the categorized batch verification. Due to these properties, A can sort out known honest and malicious messages and perform a verification process faster.

8.2 Preliminaries of Protocol 2

This section describes basic parties, the communication pattern, requirements and the cryptography background of Protocol 2.

8.2.1 Parties in Proposed Solution

The solution consists of a Trusted Authority (TA), a Group Manager (GM), a user (U) and a Vehicle (V).

- **TA** issues certified member pseudonyms and generates all public cryptographic parameters in the solution. TA is a fully trusted entity in the model and can reveal the real identity of a member (ID) in the revocation phase. TA is securely connected with all group managers (e.g. via Transport Layer Security) and manages the registration of all members.

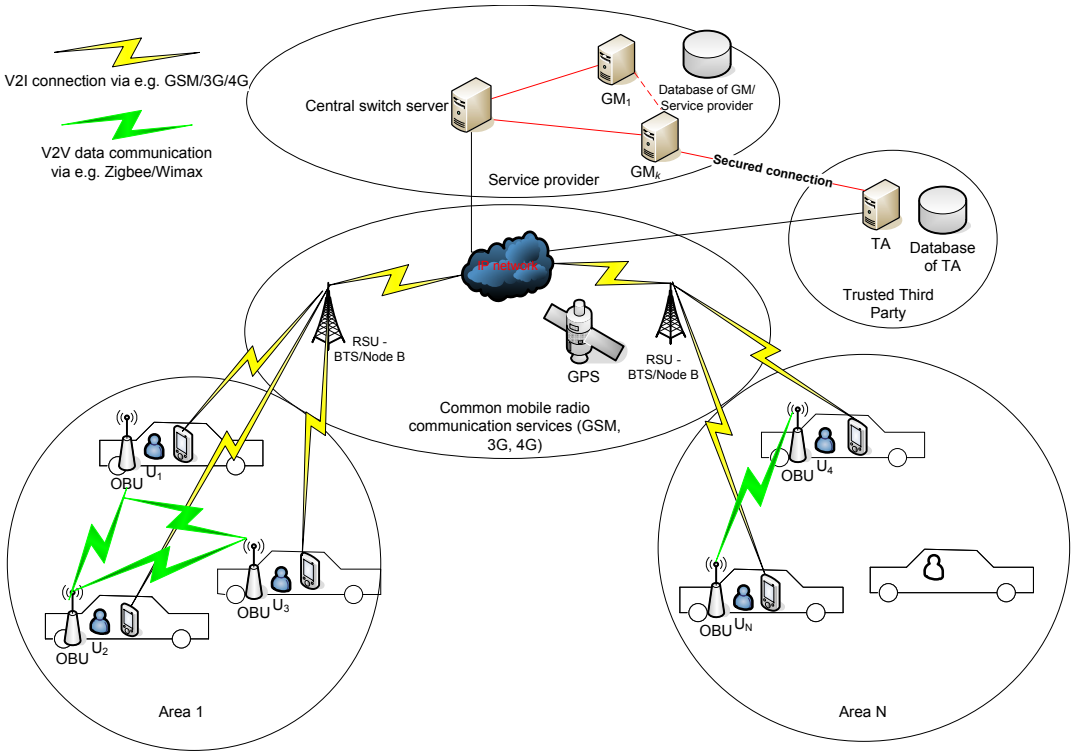


Fig. 8.2: Communication Pattern of Proposed Solution.

- **GM** is an entity which generates group secret keys to members in the join phase. In the solution, it is assumed that GM is managed by a service provider. GM broadcasts messages in the I2V communication. These messages are signed by GM. GM can also trace and open the malicious messages in its own area but it cannot reveal the user ID.
- **U** is a user with ID. After the registration of the user in TA, U obtains the certified pseudonym. Then, U can join the VANET with a vehicle. Furthermore, U can report a bogus message through the V2I communication to GM.
- **V** is a vehicle representing a user (driver) and user devices (e.g. smartphones, navigations, vehicle's OBU, ...). After joining the GM's area through V2I communication, the vehicle can broadcast and receive messages through the V2V communication or V2I-I2V communication. These messages are signed by a group signature key and verified by a batch or simple verification.

8.2.2 Communication Pattern

In this communication pattern (see Figure 8.2), a user U (specifically his/her vehicle V) can broadcast signed messages to other users/vehicles by inter-vehicle communication V2V using short/medium distance communication technologies e.g. Wimax,

ZigBee IEEE 802.15.4 or Bluetooth IEEE 802.15.1, more details can be found in [176]. It is assumed that the user owns an On Board Unit (OBU) ensuring mainly wireless communication in the V2V connection. The electronic element used to process data and interact with OBU can be an external user personal device such as a smartphone or a navigation device. These devices usually have enough computational power for basic modular arithmetic, pairing and cryptographic operations. The use of these elements and devices reduces the overall costs of the VANET architecture.

Furthermore, U can send signed messages via infrastructure connection V2I, ensuring a long-distance mobile radio communication technology e.g. GSM, 3G/4G mobile networks using Internet connection IPv4/IPv6. Road Side Units (RSU) are substituted by existing Base Transceiver Stations (BTS) in GSM or nodes B in 3G networks. Several VANET applications operating with long distances, e.g. monitoring traffic congestion or accidents, send signed messages via a V2I-I2V connection. For better efficiency of the V2I-I2V connection and fast switching of areas, the mechanisms of data aggregation and data dissemation, described in [163], can be adapted into a central switch server. These mechanisms are ensured by a service provider that issues VANET applications and navigation services. The service provider manages several group managers for specific areas. GMs are securely connected to a shared database. GMs may act as routers for incoming messages transmitted via the V2I-I2V connection. Every GM is able to verify messages received via the V2I connection while maintaining user privacy. Then, GMs send these messages to vehicles in certain areas. These messages can be signed by a GM private key and easily verified by a GM public key.

Every GM controls a specific area and releases one group public key (gpk) for this area. If a vehicle crosses different boundaries and receives messages from the neighboring area, then the vehicle determines which messages are sent from a neighboring area due to the fingerprint of gpk in these messages. The vehicle can use the group public key of the neighboring area that is stored in a device memory. The group public keys of the area and neighboring areas are obtained if the vehicle enters a new area.

8.2.3 Requirements

The proposed protocol is designed to satisfy the following security and practical requirements:

- **Privacy (Revocable Anonymity)**. The protocol protects driver's privacy in a long term. An honest driver U with a VANET device and OBU can use the pseudonym signed by TA to obtain group parameters and keys from GM. Then,

its OBU can sign every message on behalf of the group members and keep drivers' anonymity. Every malicious driver can be revealed by the collaboration of GM and TA. If some member breaks the rules, his/her messages can be opened by GM and his/her pseudonym is sent to TA, which can extract the member's ID. Next time, when an adversary requests a new pseudonym with a fresh time stamp (e.g. via IETF RFC 3161), TA checks if his/her ID appears in the list of globally revoked members.

- **Non-repudiation, Message Integrity and Authenticity**. In the V2V communication, the group signature ensures that a message is signed by a vehicle which holds the right and fresh group key pair (authenticity). The system must verify the received messages, i.e., the messages that have not been modified once they have been sent (integrity). Members stay private but can not deny that they created the signed messages (non-repudiation).
- **Short-term Linkability**. In several VANET applications like the safe changing of road lanes and the short-term mapping of vehicle movements, the short-term linkability is a desirable property [157]. In a short period, i.e., every 100÷300 ms, broadcasted V2V beacon messages are used to trace the vehicle's position and direction. The current proposals which use group signatures cannot link related messages from one vehicle sent in a short interval. Protocol 2 balances the privacy of drivers and the linkability of messages, which is available only for a short interval. On the other hand, long-term unlinkability is ensured by using the probabilistic encryption and by changing the pseudonyms in the group signature, e.g., in the V2I-I2V communication.

8.2.4 Cryptography Background

Protocol 2 employs the Elliptic Curve Digital Signature Algorithm (ECDSA) [93] as a signature scheme with the public/private keys of TA, GM, V. Petit and Mammeri [140] investigate the authentication algorithm ECDSA in vehicular networks, and processing delay of verification takes around 5 ms for ECDSA with P-256 bit curves measured on a Pentium D 3.4GHz workstation. Additionally, a probabilistic ElGamal encryption/decryption is used during the join of members. The modified short group signature WLZ scheme [169], based on the BBS04 scheme [20] is used in the V2V communication. This scheme uses bilinear maps and it is based on the q-SDH problem and the Decision Linear problem, which have been studied in [20]. These problems are described in Section 4.2.

Tab. 8.1: Notation Used in Protocol 2.

A_i	the part of a member secret key	α	a random element $\in Z_p^*$
β	a random element $\in Z_p^*$	c	a hash value in the group signature / self-challenge $c \xleftarrow{R} Z_q$
cer_{U_i}	users' certificate signed by TA	δ	a commitment value in a signature
$e()$	a pairing operation	$enc_{pk_{TA}}$	a ElGamal encryption by TA
$enc_{pk_{U_i}}$	a ElGamal encryption by U	f	the fingerprint of a group public key
g_1	a generator of G_1	g_2	a generator of G_2
G_1	a multiplicative cyclic group of a prime order p	G_2	a multiplicative cyclic group of a prime order p
$gmsk_{GM_k}$	a group manager secret key	gpk_{GM_k}	a group public key
GRL	Global Revocation List	gsk_{V_i}	a group member secret key
GTRL	Group Temporary Revocation List	γ	a random element $\in Z_p^*$
h	a random element $\in G_1^*$	H	a hash function
ch	a challenge $c \xleftarrow{R} Z_q$	ID_{U_i}	a user ID
k	a counter value	l	the security length of parameters
M	a message	μ	a commitment value in a signature
π_{U_i}	the user certificate issued by TA	p_i	a temporary result of the pairing
pk_{GM_k}	an ElGamal public key of GM	pk_{TA}	an ElGamal private key of TA
pk_{U_i}	an ElGamal private key of a user	r	random elements $\in Z_p^*$
R_i	a commitment value in a signature	s	elements in signature$\in Z_q$
sig_{GM_k}	an ECDSA private key of GM	sig_{TA}	an ECDSA private key of TA
sk_{GM_k}	an ElGamal private key of GM	sk_{TA}	an ElGamal private key of TA
sig_{U_i}	an ECDSA private key of a user	sk_{U_i}	an ElGamal private key of a user
σ	the product of a group signature	T_i	pseudonyms in a signature
TL	Temporary List	T_l	a time stamp
θ	random elements $\in Z_p$	u	the element of a group public key
v	the element of a group public key	ver_{GM_k}	an ECDSA public key of GM
ver_{TA}	an ECDSA public key of TA	ver_{U_i}	an ECDSA public key of a user
w	the element of a group public key	W	a validity value
x_i	the element of a group member secret key	Z_p	the (set of) p-adic integers
Z_q	the (set of) q-adic integers	-	-

8.3 Description of Protocol 2

This section describes the phases of Protocol 2. The notation used is described in Table 8.1. The solution focuses on the practical registration and join of VANET members and the efficient signing/verification of V2V and V2I-I2V messages. The protocol consists of seven phases: Setup, Registration, Join, Signing, Categorized Verification, Trace, Revocation.

8.3.1 Setup Phase of Protocol 2

In the setup phase $\mathbf{Set}(0,1)^l \rightarrow parameters$, TA chooses parameters $(G_1, G_2, g_1, g_2, \psi, e)$ and generates an ECDSA key pair sig_{TA}/ver_{TA}, an ElGamal private key sk_{TA} and a public key pk_{TA}. It then releases the public keys and parameters. GMs generate group signature keys, ElGamal private sk_{GM_k}, an ECDSA key pair sig_{GM}/ver_{GM} and public pk_{GM_k} keys for the secure V2I communication and publish public keys. Every GM_k randomly selects $r_1, r_2 \in Z_p^*, h \in G_1^*$ and sets $u, v \in G_1^*$ such that $u^{r_1} = v^{r_2} = h$. Then, GM_k selects random $\gamma \in Z_p^*$ and computes $w = g_2^{\gamma}$. The group public key is $gpk_{GM_k} = (g_1, g_2, u, v, w, h)$ and the group manager secret key is $gmsk_{GM_k} = (r_1, r_2)$.

8.3.2 Registration Phase of Protocol 2

In the registration phase $\mathbf{Reg}(ID_{Ui}) \rightarrow \pi_{Ui}$, the i-th user (member) U_i using a vehicle V_i with OBU requests a valid certified pseudonym π_{U_i} from TA. First, the user follows an off-line registration step to get the signed certificate cer_{U_i}. After this process, U_i owns her cer_{U_i} and he/she can perform the on-line registration step to get her pseudonym, which has an expiration time.

Off-line Registration

For the first time, TA must physically verify the driver's real ID, his/her driving license and OBU's ID number. U_i then creates an ECDSA key pair sig_{U_i}/ver_{U_i}, gives the public key to TA, which stores (ID_{U_i}, ver_{U_i}) in the database, and the signed certificate $cer_{U_i} = sig_{TA}(ID_{U_i}, ver_{U_i})$ is given to V_i.

On-line Registration

After a successful off-line registration process, the driver can request his/her pseudonym online. Assuming that U_i has pk_{TA}, ver_{TA}, the two-message of the registration phase consists of these steps:

1. U_i self-generates ElGamal key pair (sk_{U_i}/pk_{U_i}) and sends the encrypted request $enc_{pk_{TA}}(pk_{U_i}||ID_{U_i}||\ sig_{U_i}(pk_{U_i}))$ to TA.

2. TA decrypts the request and checks if ID_{U_i} is not revoked in Global Revocation List (GRL), the certificate cer_{U_i} and the user's signature, which ensures user's authenticity and commits the pk_{U_i} in the certificate with a new ElGamal key pair. Then, TA generates a challenge $ch \overset{R}{\leftarrow} Z_q$, a time stamp T_l and sends the encrypted response $enc_{pk_{U_i}}(enc_{pk_{TA}}(ID||\ ver_{U_i}||ch)||T_l||\ sig_{TA}(T_l||enc_{pk_{TA}}(ID||\ ver_{U_i}||ch)||pk_{U_i}))$ back to U_i. Finally, U_i checks the signature by TA and composes the pseudonym $\pi_{U_i} = pk_{U_i}||enc_{pk_{TA}}(ID||ver_{U_i}||ch)||T_l||\ sig_{TA}(T_l||enc_{pk_{TA}}(ID||ver_{U_i}||\ ch)||pk_{U_i})$ and stores it.

8.3.3 Join Phase of Protocol 2

In the Join phase $\mathbf{Join}(\pi_{U_i}) \rightarrow gsk_{V_i}, gpk_{GM_k}$, a vehicle V_i with the user U_i entering the k-th GM_k area for the first time, requests the group public key and his/her group member secret key. Let $H()$ be a hash function and let the two-message join phase consist of these steps:

1. V_i sends $\pi_{U_i} = pk_{U_i}||enc_{pk_{TA}}(ID||ver_{U_i}||ch)||T_l||\ sig_{TA}(T_l||enc_{pk_{TA}}(ID||ver_{U_i}||ch)\ ||pk_{U_i})$,which is encrypted using pk_{GM_k}, to GM_k.

2. GM_k decrypts π_{U_i} using sk_{GM_k}, verifies π_{U_i}, which is signed by TA and controls if $enc_{pk_{TA}}(ID||ver_{U_i}||ch)$ is not in the Group Temporary Revocation List (GTRL) and the validity of the time stamp T_l. If π_{U_i} is fine, GM creates $gsk_{V_i} = (x_i, A_i)$, where $x_i = H(enc_{pk_{TA}}(ID||\ ver_{U_i}||ch)||T_l||\gamma), A_i = g_1^{\frac{1}{x_i+\gamma}}$, and stores $(enc_{pk_{TA}}(ID||\ ver_{U_i}||ch), A_i, T_l)$ to the join table and sends ver_{GM_k}, gpk_{GM_k}, the group public keys of neighboring areas and gsk_{V_i} encrypted using pk_{U_i} to V_i.

To be noted that ElGamal encryption/decryption is probabilistic. Due to this fact, an observer can not link two or more encrypted messages if V_i requests gsk_{V_i} for the second time.

8.3.4 Signing Phase of Protocol 2

The Signing phase $\mathbf{Sig}(M, gsk_{V_i}, gpk_{GM_k}) \rightarrow \sigma$ applies the modified short group signature WLZ scheme [169], which is based on the BBS04 scheme [20]. A counter k is included in the OBUs, a member secret key $gsk_{V_i} = (x_i, A_i)$ and a group public key $gpk_{GM_k} = (g_1, g_2, h, u, v, w)$. OBU signs a message $M \in \{0,1\}^*$ and outputs the signature of knowledge $\sigma = (T_1, T_2, T_3,\ R_2, R_3, R_5, c,\ s_\alpha, s_\beta, s_x, s_\delta, s_\mu)$.
If $k = 0$, V_i generates $\alpha, \beta, r_\alpha, r_\beta, r_x, r_\delta, r_\mu \in Z_p^*$, and

computes

$$\begin{aligned} T_1 \;=\; u^{\alpha}, T_2 \;=\; v^{\beta}, T_3 \;=\; A_i h^{\alpha+\beta}, \\ \delta \;=\; \alpha x, \mu \;=\; \beta x, \end{aligned} \tag{8.3}$$

$$p_1 \;=\; e(T_3, g_2), p_2 \;=\; e(h, w), p_3 \;=\; e(h, g_2), \tag{8.4}$$

stores $T_1, T_2, T_3, \delta, \mu, p_1, p_2, p_3$, and computes

$$\begin{aligned} R_1 = u^{r_\alpha}, R_2 = v^{r_\beta}, R_3 \;=\; p_1^{r_x} \cdot p_2^{-r_\alpha - r_\beta} \cdot p_3^{-r_\delta - r_\mu}, \\ R_4 = T_1^{r_x} u^{-r_\delta}, R_5 = T_2^{r_x} v^{-r_\mu}, \end{aligned} \tag{8.5}$$

$$c = H(M, T_1, T_2, T_3, R_1, R_2, R_3, R_4, R_5), \tag{8.6}$$

$$\begin{aligned} s_\alpha = r_\alpha + c\alpha, s_\beta = r_\beta + c\beta, s_x = r_x + cx, \\ s_\delta = r_\delta + c\delta, s_\mu = r_\mu + c\mu. \end{aligned} \tag{8.7}$$

Finally, V_i increases the counter $k{+}{+}$, computes the fingerprint f of the group public key by the hash function (e.i. SHA-256) and sends the message M with the signature $\sigma = (T_1, T_2, T_3, R_2, R_3, R_5, c, s_\alpha, s_\beta, s_x, s_\delta, s_\mu, f)$.

If α and β are unchanged every n messages, the short-term linkability is kept because the pseudonyms of group signature T_1, T_2, T_3 are also unchanged. Thus, for n messages, when $1 \leq k \leq n-1$, V_i does not need to compute Equations 8.3, 8.4, contrary the WLZ scheme, but only generates random $r_\alpha, r_\beta, r_x, r_\delta, r_\mu \in Z_p^*$ and computes Equations 8.5, 8.6 and 8.7. This reduces all 3 bilinear operations to 0, 10 exponentiations to 9, and 14 multiplications to 9. This mode is suitable for the fast V2V communication where the short-term linkability is demanded. The concrete VANET application can decide when to fix the counter $k = 0$ and V_i generates new α and β and recomputes the equations 8.3 and 8.4. This mode is suitable for the V2I or V2I-I2V communication, where user privacy is more imported than the efficiency of signing. It is worth mentioning that pairing equations p_2, p_3 are fixed and can be precomputed only once.

8.3.5 Categorized Verification Phase of Protocol 2

The protocol uses a categorized verification which sorts the incoming signed messages to three levels of credibility. Due to the short-term linkability, V_i can keep the Temporary List (TL) of known vehicles. Firstly, the received message M_j is checked by V_i if it contains a valid time stamp, real and consistent data. The precise value of the time stamp, or a time window, depends on a concrete VANET application, used communication technology, distance with specific latency etc. Furthermore, V_i has to check the fingerprint f of the group public key in every received signature so that all received signed messages are from one area with gpk_{GM_k}. Received messages with

signatures that contain different fingerprints f have to be verified by the different and appropriate group public keys.

After that, the message with the group signature containing T_3 is checked if T_3 is in TL. If it holds, the recorded T_3 with previous validity (W=1) is included and sorted in the first batch. The validity W can be a boolean value which indicates valid (W=1) or invalid (and unknown, W=0) signatures. If T_3 is not in TL, the signed message with the unknown T_3 is sorted to the second batch which is verified after the first batch verification. This category is formed by the messages sent via the V2I-I2V communication. If OBU has enough time for message validation, the rest of signed messages with T_3 linked with W=0 are verified in the third batch at the end of verification. This behaviour limits the effectiveness of Denial of Service attacks where malicious cars try to use eavesdropped T_3 and generate a lot of invalid signatures with known T_3. This approach improves the efficiency of the batch verification process and helps when an attacker, who is out of the group, generates unsigned or corrupted messages.

Batch Verification

Batch verification $\mathbf{Ver}(M_j, gpk_{GM_k}, \sigma_j) \rightarrow$ valid/invalid is investigated in [66], and it verifies n messages in one batch. V_i uses $gpk_{GM_k} = (g_1, g_2, h, u, v, w)$ to verify messages $\sigma_j = (T_{j1}, T_{j2}, T_{j3}, R_{j2}, R_{j3}, R_{j5}, c_j, s_{j\alpha}, s_{j\beta}, s_{jx}, s_{j\delta}, s_{j\mu})$ for $j = 1, ..., n$.

V_i restores $\overline{R}_{j1} = u^{s_{j\alpha}} T_{j1}^{-c}$, $\overline{R}_{j4} = u^{-s_{j\delta}} T_{j1}^{s_x}$, computes a new control hash c'_j from received parameters $c'_j = H(M_j, T_{j1}, T_{j2}, T_{j3}, \overline{R}_{j1}, R_{j2}, R_{j3}, \overline{R}_{j4}, R_{j5})$,
and checks if $c'_j = c_j$. If yes, then V_i continues with verification. Otherwise, the message with the signature is inconsistent and it is refused.

V_i randomly selects $\theta_1, \theta_2, ..., \theta_n \in Z_p$ with l_b bit (the Small Exponent Test [14]), checks batch if

$$\prod_{j=1}^{j=n} R_{j3}^{\theta_j} = e(\prod_{j=1}^{j=n} (T_{j3}^{s_{jx}} h^{-s_{j\delta} - s_{j\mu}} g_1^{-c_j})^{\theta_j}, g_2) \\ e(\prod_{j=1}^{j=n} (T_{j3}^{c_j} h^{-s_{j\alpha} - s_{j\beta}})^{\theta_j}, w) \tag{8.8}$$

and if

$$1_{G_1} = (R_{j5} R_{j2})^{-\theta_j} T_{j2}^{\theta_j s_{jx} - \theta_j c_j} v^{(s_{j\beta} - s_{j\mu})\theta_j}. \tag{8.9}$$

The signed message is valid if Equations 8.8 and 8.9 hold. All T_3s from new valid signed messages are added to TL with W=1. In case that the batch verification fails, the divide-and-conquer approach is used to identify the invalid signatures that were added to TL with W=0. The honest messages keep the mark W=1.

Individual Verification

The Individual Verification phase $\mathbf{Ver}(M, gpk_{GM_k}, \sigma) \rightarrow$ valid/invalid is used at the end of the divide-and-conquer approach where the final two messages are individually verified.
V_i restores $\overline{R}_1 = u^{s_\alpha} T_1^{-c}, \overline{R}_4 = u^{-s_\delta} T_1^{s_x}$, computes new control hash c' from received parameters $c' = H(M, T_1, T_2, T_3, \overline{R}_1, R_2, R_3, \overline{R}_4, R_5)$, and checks if $c' = c$. If it is equal, V_i then continues with the verification. Otherwise, the message is inconsistent and it is refused.

Then, V_i checks if

$$R_3 = e(T_3, g_2)^{s_x} e(h, w)^{(-s_\alpha - s_\beta)} e(h, g_2)^{(-s_\delta - s_\mu)} (e(T_3, w) e(g_1, g_2)^{-1})^c \tag{8.10}$$

and

$$1_{G_1} = (R_5 R_2)^{-1} T_2^{s_x - cx} v^{(s_\beta - s_\mu)}. \tag{8.11}$$

The signed message is valid if equations 8.10 and 8.11 hold.

It is obvious from Equations 8.8 and 8.10 that the individual verification has a cost of 5 pairing operations per one message but the batch verification costs only 2 pairing operations per n messages. This is the main reason the individual verification is not used and it is proposed to use the categorized batch verification.

In some long-distance VANET applications, GMs may act as routers for incoming messages transmitted via V2I-I2V communication. In this case, GM_k receives the messages and verifies their signatures, signs the valid ones using its own private ECDSA key sig_{GM_k}, and finally submits them to all the users in a certain k-area. Then, these users can easily verify the signature issued by GM_k using the public ECDSA key ver_{GM_k}.

8.3.6 Trace Phase of Protocol 2

In Trace phase $\mathbf{Trace}(M, \sigma, gmsk_{GM_k}) \rightarrow gsk_{V_i}, \pi_{U_i}$ every bogus signed message can be opened by GM_k using the group manager secret key $gmsk_{GM_k} = (r_1, r_2)$. Bogus messages are messages with correct signatures that carry malicious content which can cause problems in traffic. GM_k extracts the part of the member secret group key $gsk_{V_i} \rightarrow A_i = T_3/(T_1^{r_1} \cdot T_2^{r_2})$ and searches the record $(enc_{pk_{TA}}(ID||ver_{V_i}||c), A_i, T_l)$ in the database. The part of the member pseudonym can be sent to TA for revocation.

8.3.7 Revocation Phase of Protocol 2

In the Revocation phase $\mathbf{Rev}(\pi_{U_i}) \rightarrow ID_{V_i}$, a member can be revoked. When there are serious circumstances, e.g., an accident, a malicious member is revoked globally by the cooperation of GM_k and TA. GM_k is able to open a message and extract the member pseudonym that is sent to TA. TA broadcasts $rev = (enc_{pk_{TA}}(ID||ver_{V_i}||c), T_l)||\ sig_{TA}(rev)$ to other active GMs which check the signature and store rev to own GTRLs until the lifetime of this pseudonym expires. TA extracts ID_{V_i} and adds it to GRL so that the malicious member can not refresh his/her pseudonym in the next registration phase.

8.4 Security Analysis of Protocol 2

In the following parts, the adversary model and the possible attacks the proposed protocol has to be robust against is detailed. These attacks are related to the security requirements which must be fulfilled by the proposed protocol, namely: *revocable anonymity*, *message integrity* and *message authenticity*.

8.4.1 Adversary Model

This attacker model considers an adversary who can control vehicles and can also access communication lines to capture, modify and retransmit messages. In this way, he/she can be a purely external attacker and also an internal one. In any case, his/her computational power does not permit the adversary to break current computationally secure cryptosystems.

Regarding the other entities of the proposed system, the Trusted Authority (TA) is managed by some governmental organization such as the traffic authority of each country. Therefore, this entity is fully trusted. Regarding the Group Managers (GMs), these elements are assumed to be managed by some company that participates in the system as a service provider. In this way, GMs are expected to follow the proposed protocol in an honest way (i.e., they will not tamper with messages, drop them, etc) but they may try to retrieve the real identities of the users who use the VANET. Gathering real identities and other personal data may report significant economical benefits to the company in charge [2] and it is an explicit privacy threat. Therefore, they are covered in the proposed adversary model as passive attackers that uniquely try to break the privacy of the legitimate users. In this way, they will not participate in any other kind of an attack. Moreover, the RSUs which are used by the GMs to communicate with the vehicles of the VANET are assumed to be tamper-proof elements which cannot be compromised by external attackers.

The attacks that can be performed by the considered adversaries are summarized in the following text. They can be broadly divided into *passive* and *active* attacks:

- *Passive attacks.* They only require the attacker to have access to the communication lines. Their main purpose is to jeopardize the privacy of the users by compromising the confidentiality and/or unlinkability of the submitted messages. Specifically, those attacks are:
  - Eavesdrop messages transmitted between V_i and TA in the *Online Registration step.*
  - Eavesdrop messages transmitted between V_i and GM in the *Join step.*
  - Eavesdrop messages transmitted between TA and GM.
  - Trace the V2V/V2I messages sent by a certain user.
  - Retrieve the real identity of a certain user in the *Join step.*
- *Active attacks.* These attacks are based on tampering with valid messages, submitting fake ones, etc. Their main purpose is to get some benefit or simply disrupt the normal execution of the proposed protocol. This kind of attacks generally compromise the integrity and/or the authenticity of the submitted messages. Specifically, the proposed protocol should be strong against:
  - Tamper with messages transmitted between V_i and TA in the *Online Registration step.*
  - Tamper with messages transmitted between V_i and GM in the *Join step.*
  - Tamper with V2V/V2I messages sent by legitimate vehicles.
  - Tamper with messages transmitted between TA and GM.
  - Generate a fake but valid pseudonym.
  - Allow unauthorized users to generate fake but valid V2V/V2I messages.
  - Launch a DoS attack against the vehicles of the VANET.
  - Reuse former messages to perform replay attacks.
  - Use the anonymity provided by the scheme to misbehave without being traced.

8.4.2 System's Behaviour against the Considered Attacks

The next part explains how the proposed protocol deals with the attacks which have been introduced above. Note that some of these attacks may be covered together in the same subsection.

Eavesdrop Messages Transmitted during the Different Steps of the Protocol

First, it is focuses on the messages transmitted between V_i and TA in the *Online Registration step.* In this case, V_i sends a request $(enc_{pk_{TA}}(pk_{U_i}||ID_{U_i}||sig_{U_i}(pk_{U_i})))$ in

order to get a new pseudonym and TA answers with a response ($enc_{pk_{U_i}}(enc_{pk_{TA}}(ID$ $||ver_{U_i}||ch)||T_l||sig_{TA}(T_l\ ||enc_{pk_{TA}}(ID||\ ver_{U_i}||ch)||pk_{U_i})))$. Both messages are encrypted using the ElGamal cryptosystem (nowadays this cryptosystem is considered to be secure [164]) and, hence, the attacker is unable to decrypt them and get the transmitted data because decryption requires the knowledge of the secret keys sk_{U_i} and sk_{TA}. These keys are only known by the legitimate user and the trusted authority, respectively.

Similarly, the attacker cannot get the data transmitted between V_i and GM in the *Join step* because these messages are also encrypted using the ElGamal cryptosystem. In this case, the secret keys that are needed to obtain the sensitive information are sk_{U_i} and sk_{GM_k}. Both keys are only known by the legitimate user and the contacted group manager. Note that, as explained previously, group managers are controlled by a service provider and, hence, they are expected to behave honestly.

Finally, the attacker cannot disclose any information from the messages sent between the trusted authority and the different group managers due to the fact that these communications are always secured using TLS.

Trace the V2V/V2I Messages Sent by a Certain User

Vehicles apply the modified short group signature WLZ scheme [169] to sign the V2V/V2I messages that they submit. Group signatures generated under this scheme contain the group members' pseudonyms T_1, T_2, T_3 which are a linear encryption of members' secret key A_i and random α and β. The short-term linkability property of the messages does not violate the drivers' privacy. When the counter k is set to 0 and V_i generates new values for α and β, the new generated signatures are unlinkable with the former ones because they contain new values for T_1, T_2 and T_3.

Retrieve the Real Identity of a Certain User in the *Join step*

This attack is based on retrieving the real identity of a certain user from its pseudonym π_{U_i} in the *Join step*. Note that this attack can only be performed by the GM that is expected to receive the message because π_{U_i} is encrypted using its ElGamal public key pk_{GM}.

Pseudonym π_{U_i} contains the identity of the user (ID) encrypted with the ElGamal public key pk_{TA}, which is only known by TA. Therefore, GM cannot retrieve the real ID. Nevertheless, GM is capable of linking all the request messages that contain the same π_{U_i}. In order to minimize this issue, the user should update π_{U_i} with a certain frequency (following the *Online Registration step*).

Tamper with Messages Transmitted during the Different Steps of the Protocol

Focusing on the messages transmitted between V_i and TA in the *Online Registration step*, message integrity and authenticity are ensured by the ECDSA signature scheme. The *request message* contains the member public key pk_{U_i} signed with the ECDSA signature key sig_{U_i}. Assuming that both the ECDSA signature scheme and the hash function in use are secure, if the request message is modified in any way, the ECDSA verification process will detect this situation.

Messages transmitted between V_i and GM in the *Join step* also ensure integrity and authenticity. First, V_i submits its pseudonym, which is signed by the TA using the ECDSA signature scheme. Then, GM sends to V_i its assigned group member secret key $gsk_{V_i} = (x_i, A_i)$. The use of a hash function to compute gsk_{V_i} together with the use of ElGamal cryptosystem to encrypt the message provide integrity and authenticity.

Regarding the V2V/V2I messages sent by the vehicles, those elements are signed and verified employing the modified short group signature WLZ scheme [169]. This approach ensures message authenticity and integrity to those messages.

Finally, the attacker cannot tamper with the data exchanged between the trusted authority and the different group managers due to the integrity and authenticity properties provided by the use of TLS.

Generate a Fake but Valid Pseudonym

If the attacker wants to create a valid pseudonym π_{U_i}, he/she needs the ECDSA private key sig_{TA}. This secret key is only known by the TA and, hence, the attacker cannot obtain it to launch this attack.

It is worth mentioning that if an illegal π_{U_i} is sent to a legitimate user, he/she can use the TA's public ECDSA key ver_{TA} to verify its validity.

Allow Unauthorized Users to Generate Fake but Valid V2V/V2I Messages

The attacker can launch this attack by signing a new fake message on behalf of a group of legitimate users or by modifying a message signed and submitted by a legitimate user.

The signing and verification phases employ short group signatures with the short-term linkability to ensure message authenticity and integrity. As explained previously, the protocol applies the modified short group signature WLZ scheme [169]

and inherits all its security features. As a result, only the group manager GM_i and the valid group members U_i can sign a message on behalf of the group.

If an attacker without the valid $gsk_{V_i} = (A_i, x_i)$ is willing to modify a certain message, he/she must recompute the hash c and some signature parts. Assuming that the hash function is secure and that the Discrete Logarithm problem holds, computing ($s_{j\alpha}$, $s_{j\beta}$, s_{jx}, $s_{j\delta}$, $s_{j\mu}$) without knowing x_i is considered unfeasible. If this proof of knowledge is incorrectly computed, equations 8.8, 8.9 and 8.10, 8.11 will not hold during the verification step.

Launch a DoS Attack against the Vehicles of the VANET

The attacker can launch a DoS attack by broadcasting a large number of bogus messages containing fake pseudonyms and signatures. This attack can be more effective if several attackers collaborate on this purpose (note that, a Sybil attack can be considered to achieve this).

As a result of this attack, legitimate users will be flooded with a large amount of messages and they will not be able to process all of them. The straightforward solution for this situation is to discard some of the received messages (or all of them). The problem of this approach is that some of these discarded messages can be legitimate warnings of some dangerous situation. In order to prevent it, the proposed protocol implements a *categorized batch verification* step.

In this way, a honest user has a Temporary List (TL) of other known and honest drivers, which uses the short-term linkability property that keeps the pseudonym T_3 of each signed message unchanged for a certain period of time. This user receives messages and checks the TL to put the messages containing a known T_3 in the first batch of verification (the one with the highest priority). Messages with an unknown pseudonym are stored in the second batch. Finally, potentially untrusted messages (e.g., with validity $W = 0$) are verified in the third batch only if verifier's OBU has free time and computational capacity to do it.

Reuse Former Messages to Perform Replay Attacks

Submitted messages contain a time stamp with current time and date. Before being verified, the time stamp of each received message is checked. If an attacker without the valid $gsk_{V_i} = (A_i, x_i)$ is willing to reuse an old message with a valid signature, he/she must refresh the time stamp and then recompute the hash c_j and the signature $(s_{j\alpha}, s_{j\beta}, s_{jx}, s_{j\delta}, s_{j\mu})$. Note that obtaining valid values for $s_{jx}, s_{j\delta}$ and $s_{j\mu}$ without knowing x_i is unfeasible under the Discrete Logarithm problem.

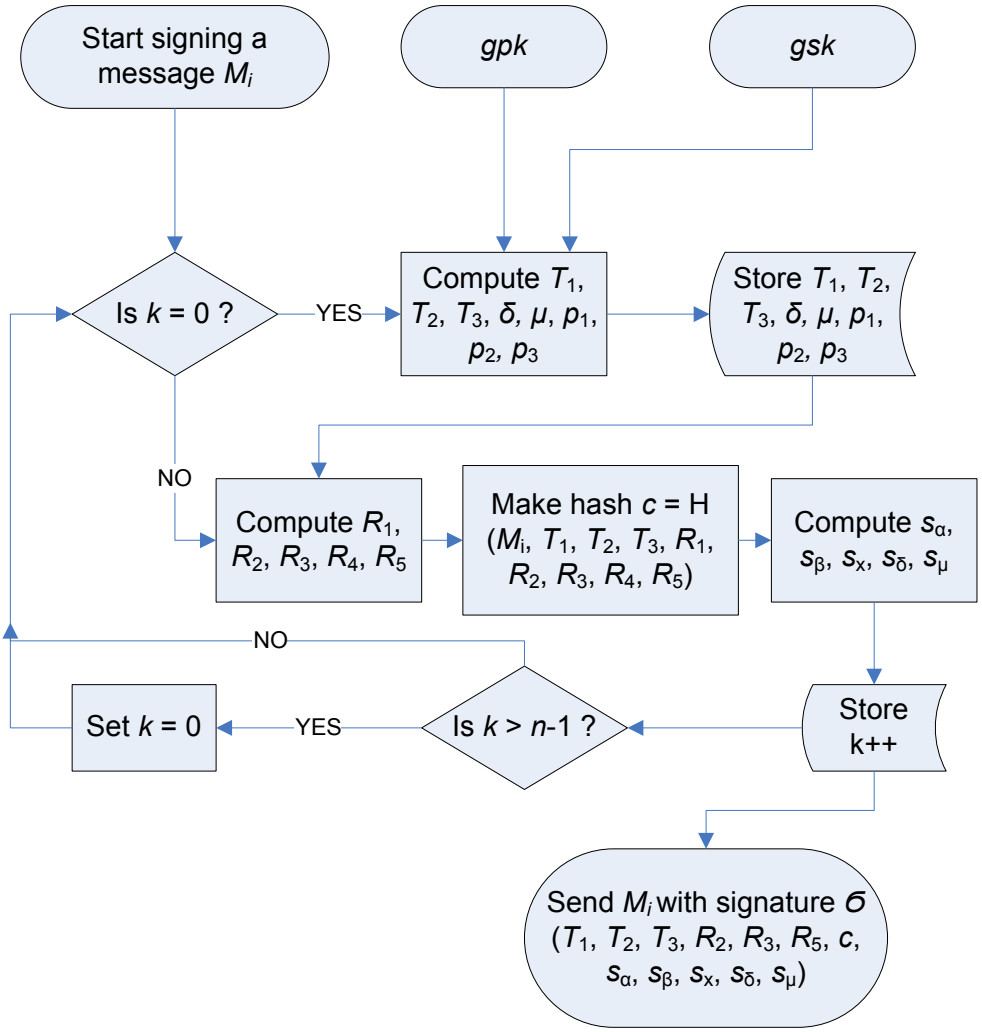


Fig. 8.3: Process Flowchart of Signing.

Use the Anonymity Provided by the Scheme to Misbehave without Being Traceable

The proposed protocol provides anonymity and unlinkability for drivers in front of other vehicles and GMs. Nevertheless, this protection can be revoked if the GM of the area and the TA collude. Since both entities are honest, this will be assumed to happen only if the driver misbehaves.

If this is the case, each correct message submitted by a malicious member can be opened by the GM using its group manager secret key $gmsk_{GM_k}$. In this way, the GM extracts the part of the member secret group key $gsk_{V_i} \rightarrow A_i = T_3/(T_1^{r_1} \cdot T_2^{r_2})$ and searches the record $(enc_{pk_{TA}}(ID||ver_{V_i}||c), A_i, T_l)$ in the database. Finally, the part of the member pseudonym can be sent to the TA for retrieving the real ID.

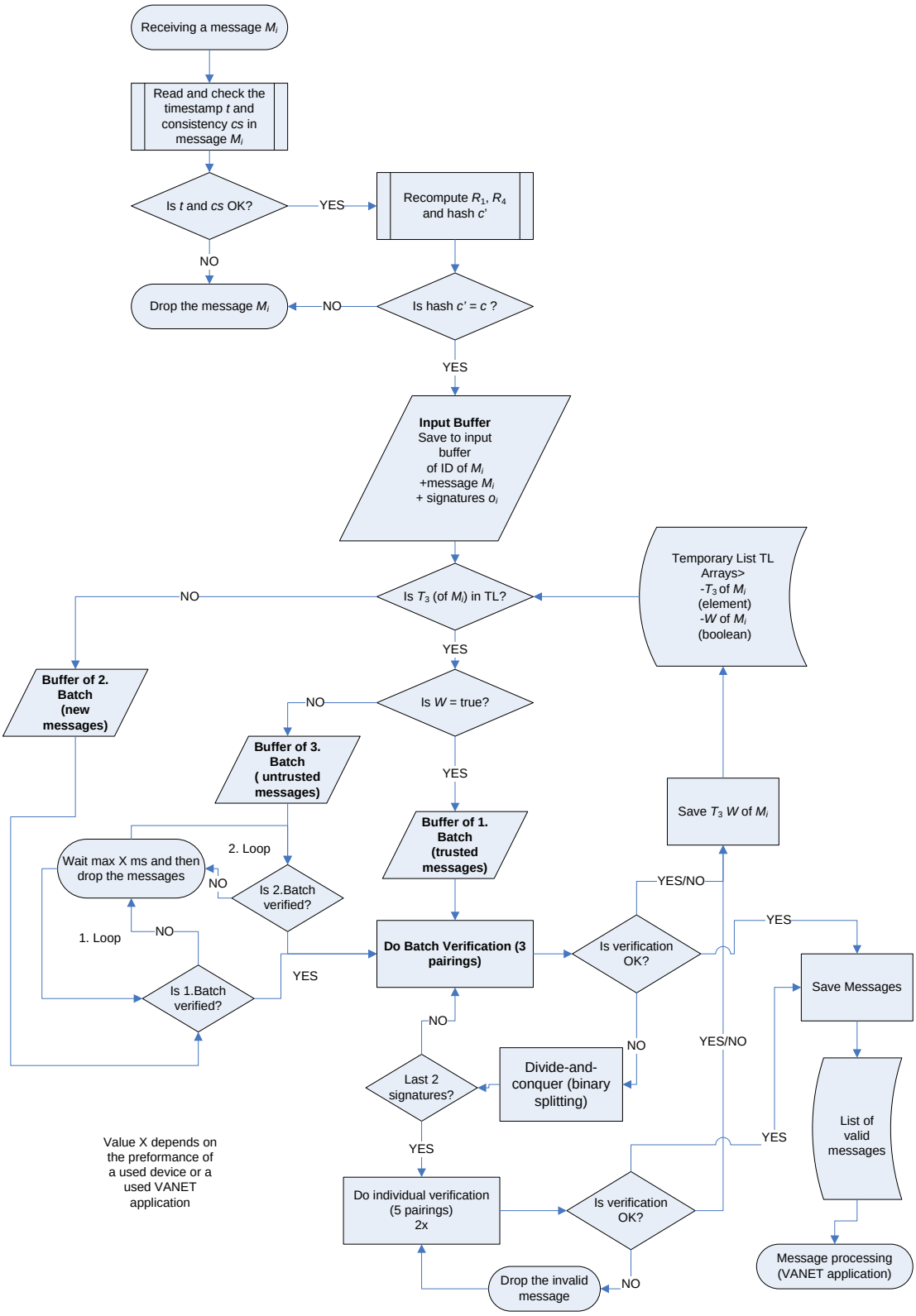


Fig. 8.4: Process Flowchart of Categorized Batch Verification.

8.5 Experimental Implementation of Protocol 2

The proposed protocol has been implemented as a proof-of-concept in JAVA (PC) and on the Android platform (smartphones). The main core of the experimental implementations is formed by the group signature scheme that uses the Java Pairing Based Cryptography (jPBC) Library[1] in both test scenarios (jPBC on Java for the PC version and wrapped jPBC on Android for the smartphone version). The implementation employs the MNT curves type D with the embedding degree $k = 6$, the 171-bit order of curves and the pre-generated parameters d840347-175-161.param.

The registration and join phases use the ECDSA signature scheme and ElGamal cryptosystem that are provided by the Bouncy castle Library [2]. All ECDSA and ElGamal keys can be inherited from class org.bouncycastle.jce.provider.JDKKeyFactory. The 1024-bit ElGamal encryption and the 256-bit ECDSA scheme with the SHA-1 hash function are employed.

In the signing phase, Fig. 8.3, a string of a message M_i, counter k, a member secret key gsk and a group public key gpk input to the signing process. There are two modes of signing: an initial signing mode and a normal signing mode. The initial mode of signing is performed if $k = 0$. The signature algorithm then computes $T_1, T_2, T_3, \delta, \mu, p_1, p_2, p_3$, where 3 pairing operations are computed. This mode is used in the V2I-I2V communication, respectively, in the long distance VANET applications.

The normal mode of signing is performed if $1 \leq k \leq n-1$ and the signature algorithm uses the stored parameters $T_1, T_2, T_3, \delta, \mu, p_1, p_2, p_3$. M_i is signed and the signature σ with elements $T_1, T_2, T_3, R_2, R_3, R_5, c, s_\alpha, s_\beta, s_x, s_\delta, s_\mu$ is produced. Then the message M_i containing the signature σ is sent. The normal mode is used in the fast V2V communication,in the short distance VANET application respectively. The proposed signing phase is depicted as a flowchart in more detail in Fig. 8.3.

A receiver (verifier) receives the M_i and checks the time stamp and consistency of the message. Then, the receiver checks the validity of elements R_1, R_4, c' and saves the incoming M_i to an input buffer. Messages are sorted out into three categories, and 3 buffers, respectively. The sorting process is based on knowing the T_3 of incoming messages and the validity indicator W (a boolean type). Depending on the permitted number of received messages and the maximal time limit of the verification phase, the verifier starts to do the batch verification.

The categorized verification process outcomes the list of valid messages and upgrades a temporary list with the elements T_3 and W. If M_i is valid, then W is set to true. Otherwise, if M_i is invalid, then W is set to false. The proposed categorized

[1](avail. on http://gas.dia.unisa.it/projects/jpbc/index.html)
[2](avail. on http://www.bouncycastle.org/resources.html)

V2V scheme:	Proposed Scheme	WLZ scheme [169]	GSIS [108]	Zhang et al. [179]	Ferrara et al. [66]
Batch:	yes	yes	no	yes	yes
Short-term linkability:	yes	no	no	no	no
Length of signature:	$5G_1, G_T, 5Z_p, f$ (2636 bits)	$5G_1, G_T, 5Z_p$ (2380 bits)	$3G_1, 6Z_p$ (1500 bits)	$7G_1, G_T, 5Z_p$ (2570 bits)	$3G_1, G_T, 6Z_p$ (2032 bits)
Performance of batch verification					
Pairings	**2**	**2**	5n	2	2
Exponentiation	**11n**	**11n**	12n	14n	13n
Multiplication	11n+1	11n+1	**8n**	17n	10n+1
Performance of individual verification					
Pairings	5	5	5	5	5
Exponentiation	**10**	**10**	12	12	12
Multiplication	9	9	8	8	8
Performance of initial mode signing / normal mode signing					
Pairings	3 / **0**	3 / 3	3 / 3	3 / 3	3 / 3
Exponentiation	12 / **9**	10 / 10	12 / 12	12 / 12	12 / 12
Multiplication	12 / **9**	14 / 14	12 / 12	12 / 12	12 / 12

Tab. 8.2: Comparison of Verification and Signing.

batch verification is depicted in more detail as a flowchart in Fig. 8.4. At the end of the verification process, the valid messages are sent to VANET applications depending on their time priority. The performance results of the implemented signing and categorized verification phases are outlined in the following section.

8.6 Evaluation of Protocol 2

This section outlines a theoretical evaluation and comparison of the proposed protocol with the related VANET schemes which use group signatures, GSIS [108], Zhang et al. [179] , Ferrara et al. [66] and the scheme of Wei et al. (WLZ scheme) [169]. This evaluation is independent from the used machine. In addition, the experimental implementation of the proposed protocol and the implementation of BBS group signature scheme used in the related works are compared. The implementation of the solution runs on two platforms, namely JAVA (PC) and the Android platform (smartphones).

8.6.1 Theoretical Evaluation and Comparison

Generally, the time of bilinear pairing T_p is considered the most expensive operation (ten times more expensive than exponentiation operation T_e) and exponentiation

is more expensive than multiplication T_m. The modular arithmetic operations like addition and subtraction can be computed more efficiently than multiplication and exponentiation as is in Chapter 6. Consequently, these fast operations in this performance evaluation are omitted. In the proposed scheme, the initial signing mode takes $3T_p + 12T_e + 12T_m$ and the normal signing mode takes only $9T_e + 9T_m$. The computation complexity of the verification is linear and depends on the number n of received messages. The verification takes $2T_p + 11nT_e + (11n + 1)T_m$ in the batch verification mode, and $5T_p + 10T_e + 9T_m$ in the individual verification mode.

The signing phase of the proposed scheme costs less exponentiations than the signing phase of the related schemes. Moreover, during the normal mode signing of x messages with short-term linkability, all operations are significantly reduced to pairing ($3 \Rightarrow 0$), exponentiation ($10 \Rightarrow 9$) and multiplication ($14 \Rightarrow 9$).

The designed protocol based on the group signature scheme BBS04 [20] reaches more efficient batch verification ($2\ T_p + 11n\ T_e$) and individual verification ($5\ T_p + 10\ T_e$) than the compared schemes. The results in Table 8.2 are not influenced by optimization techniques apart from the batch verification. But the related solutions like Zhang et al. [179], Ferrara et al. [66], the WS2010 scheme [168] and also the WLZ scheme [169] use uncategorized batch verification that can be negatively affected by malicious messages ($\geq$ 15% from all messages). It is assumed that the proposed protocol applies the categorized batch verification with the short-term linkability in VANET for the first time. The categorized batch verification with the temporary list of known vehicles reaches the high correctness of the important first batch in case the bogus or damaged signed messages appear in the V2V communication. In case a malicious driver Eve (E) starts the Sybil attack, which is a special kind of the DoS attack, then she broadcasts bogus messages that contain fake pseudonyms and signatures. Meanwhile, the honest drivers (C, D, F,...) send messages that contain valid pseudonyms and signatures announcing an accident (sent by D) or a traffic jam (sent by C). If existing solutions are used, E can flood the uncategorized batch verification process and paralyze drivers who must discard some messages.

The proposed solution uses categorized batch verification. Driver Bob (B) has a Temporary List (TL) of honest drivers. It is supposed that Bob's TL keeps the list of known and honest drivers like D, F,... using the property of the short-term linkability, which keeps the pseudonym T_3 unchanged for a short time (depends on VANET applications). If B receives all messages, he checks the TL and collects the messages containing known T_3 to the first batch, and then B verifies them. Therefore, the warning message referencing the accident from driver D is verified in time. The messages with unknown pseudonyms like those from driver C are collected to the second batch. The potentially untrusted messages from driver E with validity W=0 are verified in the third batch only if Bob's OBU has free time

and computational capacity for this. If Eve tries to replay recent valid pseudonyms together with false signatures, then the recomputed hash c'_j is not equal to received hash c_j due to time stamps in messages. For this reason, Eve is not able to mount a successful DoS attack against the batch verification of signatures.

8.6.2 Practical Comparison and Results

The JAVA implementation of the solution has been tested on a PC with Intel(R) Xeon(R) CPU X3440 @ 2.53GHz, 4 GB Ram, Windows 7 Professional. The Android implementation has been tested on two smartphones: Google Nexus S with CPU Cortex-A8 @ 1 GHz and 512 MB Ram, and Samsung Galaxy S3 with CPU 4xCortex-A9 @ 1.4 GHz and 1024 MB Ram.

Results of JAVA Implementation

The signing phase of the scheme is compared with the BBS scheme [20], which is also used in the GSIS scheme [108] & Zhang *et al.* scheme[179] & Ferrara *et al.* scheme [66] and the scheme of Wei *et al.* [169] (see Figures 8.5 and 8.6). The normal mode of the proposed signing phase takes approximately 55 ms per 1 signing. This is more efficient than the compared schemes based on the BBS scheme [20] taking approx. 165 ms per 1 signing, because the proposed solution reduced 3 pairing operations to 0 pairings for n messages in the signing phase. The initial mode of the designed signing phase takes approx. 165 ms due to the same number of operation as BBS scheme. This slowed mode is used in the long distance VANET applications where the privacy must be kept and the time of data processing is not critical.

The performance of the Verification phase in the proposed protocol is more efficient than related BBS schemes (see Figures 8.7 and 8.8). The verification of a single signature takes approx. 207 ms using the proposed scheme, and approx. 224 ms using related schemes based on the BBS04 scheme. Figure 8.8 demonstrates the efficiency of the batch verification. If the batch verification is employed, then the verification of one signature takes only approx. 50 ms on average so the batch verification of 10 signatures takes approx. 500 ms. Then, the verification of 6 signed messages takes approx. 300 ms. In the short distance VANET applications, e.g. break alerts, the vehicle controls the nearest vehicles only in the front of its direction. With the measured numbers and used hardware, the protocol can monitor and verify the 6 signed messages from 6 vehicles that are in front of the receiving vehicle. Assuming that the device that supports optimized cryptographic operations like exponentiation, multiplication and pairings is used, then the protocol is able to monitor and verify tens of cars in close distances. Moreover, due to the short-linkability, the receiver sorts out the known and potentially honest signed messages.

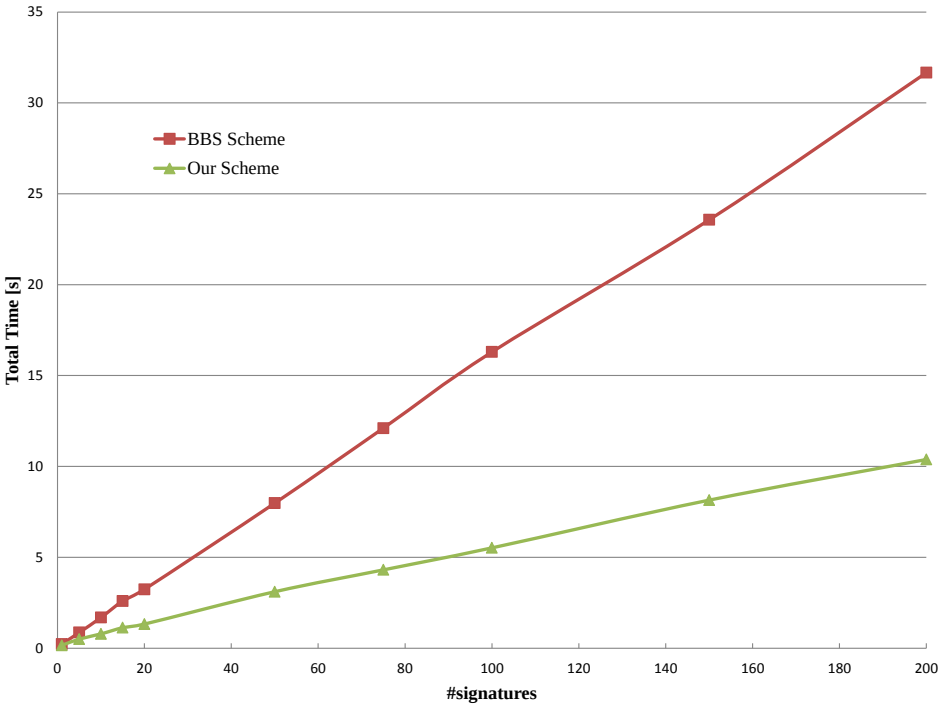


Fig. 8.5: Performance of Signing Phase on PC Machine.

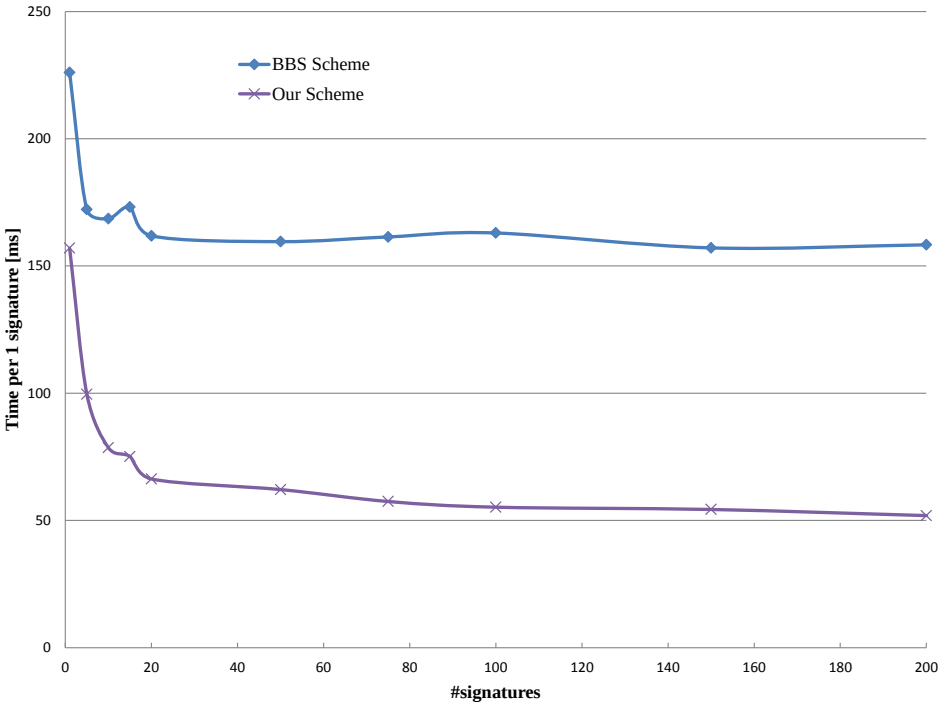


Fig. 8.6: Performance of Signing Phase (per 1 Signature) on PC Machine.

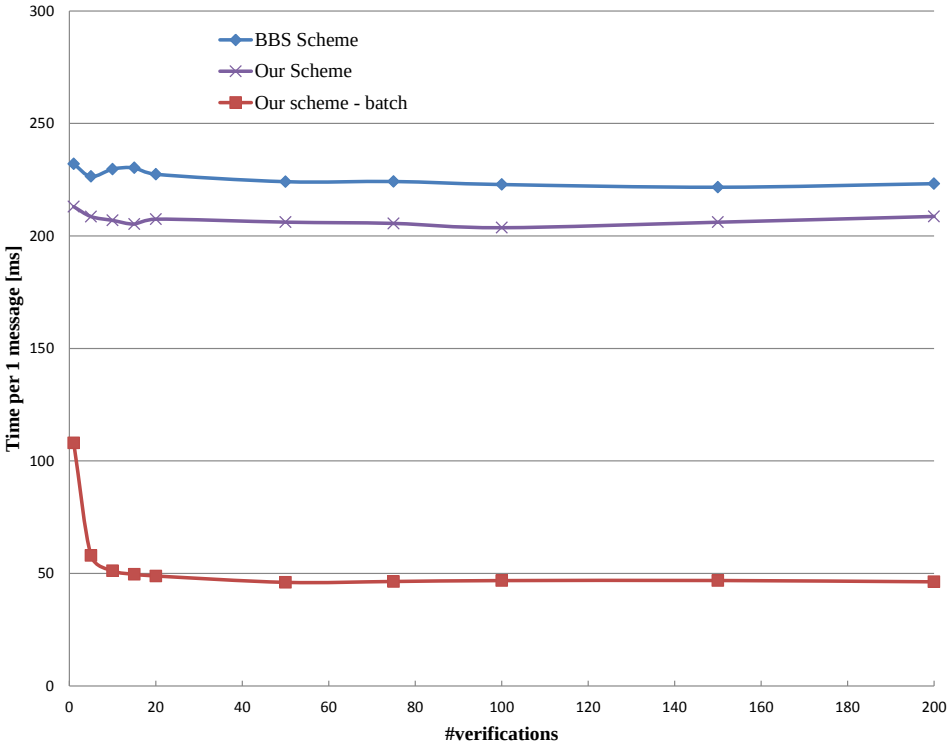


Fig. 8.7: Performance of Verification Phase (per 1 Verification) on PC Machine.

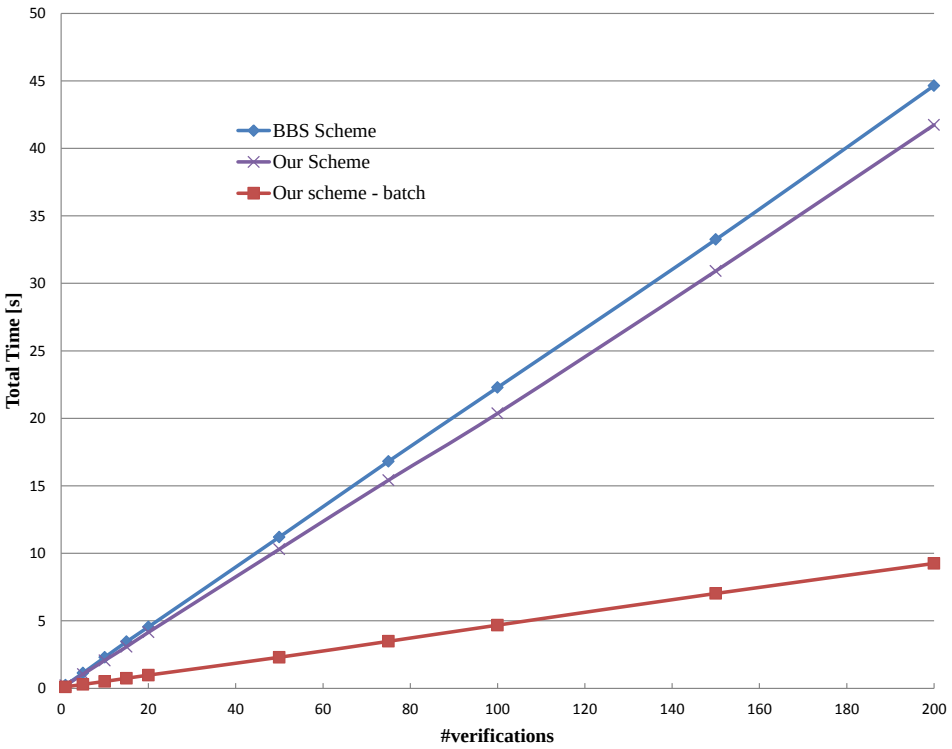


Fig. 8.8: Performance of Verification Phase on PC Machine.

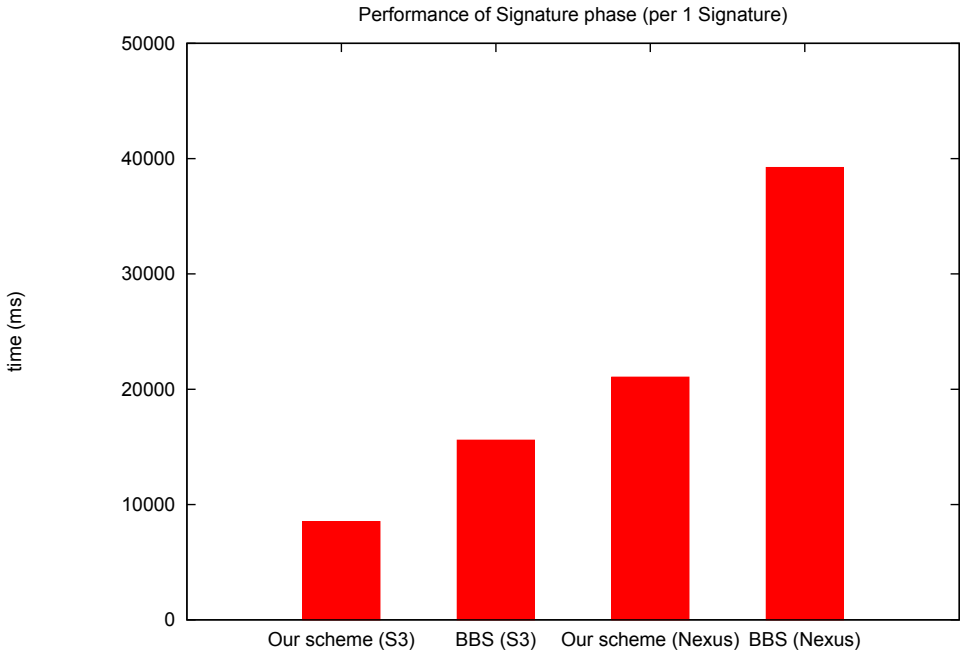


Fig. 8.9: Performance of Signing Phase (per 1 Signature) on Smartphones.

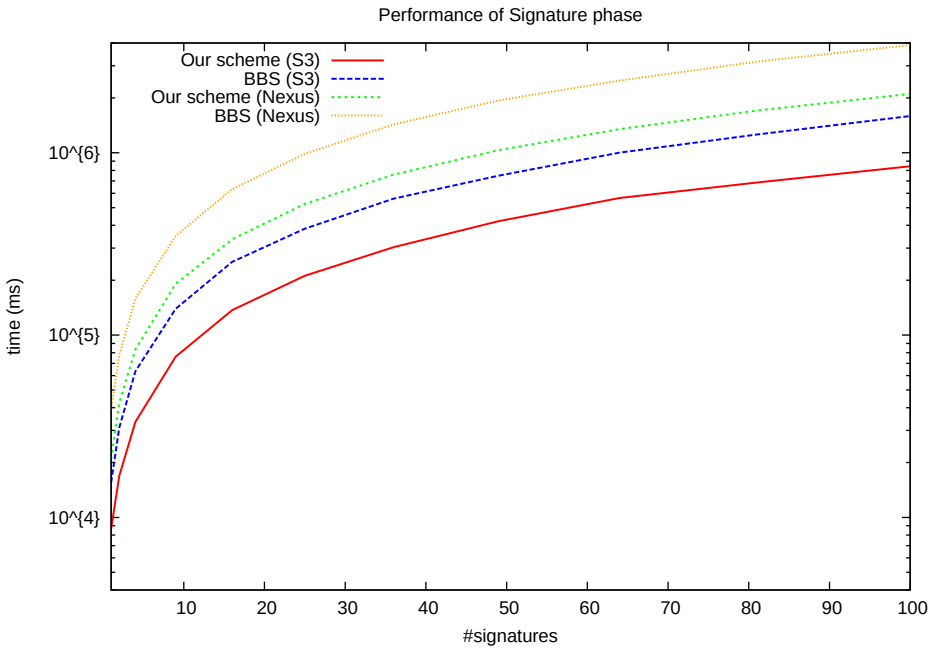


Fig. 8.10: Performance of Signing Phase on Smartphones.

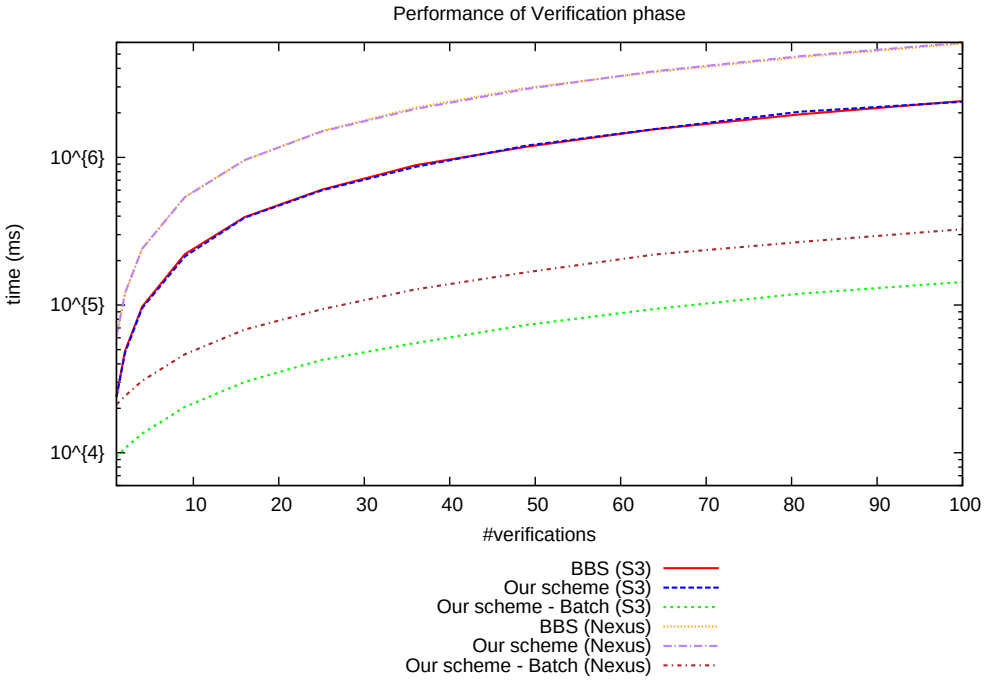


Fig. 8.11: Performance of Verification Phase on Smartphones.

Results of Android Implementation

Moreover, the proposed protocol have been tested on two smart-phones, Google Nexus S and Samsung Galaxy S3, which use the Android platform and support the jPBC Library. Figures 8.9, 8.10 and 8.11 show their performance when signing and verifying messages. These results reflect that the solution can effectively monitor events related to long distance VANET applications, such as traffic jams, accidents, on-road weather reports etc. Note that these messages are transmitted via V2I-I2V connection.

Furthermore, a remarkable difference can be observed between the execution time achieved by the two smartphones (Nexus S and Galaxy S3). The newer device (Galaxy S3) has more computation power and, hence, it computes all the operations faster. This is especially helpful to perform the signing step in a realistic period of time and, hence, enable the proposal to be deployed in real environments. Regarding the batch verification step, although a reasonable performance is obtained at this point, this step should be executed in the OBUs for practicability.

It should be stressed that the Divide-and-conquer process can be used in those cases when the batch verification fails due to the presence of a fake message. Therefore, this algorithm can be used to split and process the messages until the fake one is found. Moreover, aggregated messages can be computed and stored to avoid re-computing them again if the verification fails. The cost of this process is logarithmic

($log_2 N$) and it still improves the cost of individual verification, which is $(N+1)/2$. Ferrara *et al.* show in [66] that the batch verification step of the BBS scheme can be more efficient if fake messages are less than 15 %. The categorized verification process which is proposed in the scheme minimizes the rate of fake messages in the first priority batch.

8.7 Summary of Chapter 8

This chapter presents a comprehensive security solution of vehicular networks that protects the driver's privacy. The proposed cryptographic protocol focuses on users' privacy while messages are transmitted between vehicles and between users and the infrastructure. It is assumed that the infrastructure is maintained by a group manager. Furthermore, the proposed protocol prevents the denial of service attacks, which are a current problem of many secure and privacy-preserving proposals in vehicular networks. The proposed verification is categorized. Thus, this categorized verification is able to detect and remove some fake messages in the first stage and the second stage processes less messages. The results of the experimental implementation on the PC point to the fact that the proposed security solution with batch verification can be used in the short distance VANET applications which demand a fast message verification. Smartphones have lower computational power than PCs, so they could be used for processing long-distance VANET applications because a small computational delay would not cause difficulties. It is assumed that GM has a greater computational power than OBU or smartphones, and it can take the responsibility of verifying the signatures transmitted via V2I-I2V communication. Moreover, the protocol is three times faster in signing than related schemes due to the short-term linkability. In long distance VANET application, the protocol keeps users' privacy, guaranteeing that nobody can create a profile of them.

9 PRIVACY-PRESERVING FRAMEWORK FOR HETEROGENEOUS NETWORK

In this chapter, the privacy preserving cryptography framework applied in heterogeneous networks is proposed. The proposed solution deals with user privacy in geosocial applications. The framework has been published as the journal paper (IF 0.311) in [117]. The solution uses the modification of the group signature scheme based on the BBS04 scheme similarly like the protocol in the previous chapter.

9.1 Privacy in Geosocial Services

Geosocial applications have become very popular but can misuse user's private data and location. The proposed solution prevents tracking and protects against personal identity and location being misused by external attackers or service providers. The proposed framework provides security and privacy protection for geosocial applications that provide, for example, information sharing, geotagging and monitoring of people without revealing their identity to unauthorized persons, including the service provider. Unlike current security solutions in geosocial services, the designed solution uses advanced cryptography to secure user privacy. This protection is provided by advanced group signatures ensuring data integrity, authenticity, non-repudiation as well as strong privacy protection. This chapter outputs a detailed cryptographic solution for the protection of privacy in geosocial services and its security analysis. The proposed solution has also been implemented and the performance results are outlined.

Emerging geosocial applications like geotagging, check-in-based services, social-mapping services or user reviewing have come to be very popular with smartphone users and can be useful in sharing information, from restaurant reviews to information about disasters, e.g. an emerging epidemic, flood level, and so on. The user position is usually used in geosocial services such as Foursquare, Yelp, Gowalla, Facebook Places and so on. Users with their devices (smartphones, mobiles, laptops, tablets,...) send and download data, broadcast notices and request the location-based information. Nevertheless, the user authentication process which uses a user identity suffers from the possibility of user identity misuse and unauthorized user profiling.

These systems, which continuously track user's location, can leak private information into wrong hands. Due to this fact, the confidence in these systems decreases. If geosocial services do not preserve privacy, there will always be the risk of misusing the system, e.g. sending targeted ads or, in the worst case, kidnapping facilitated

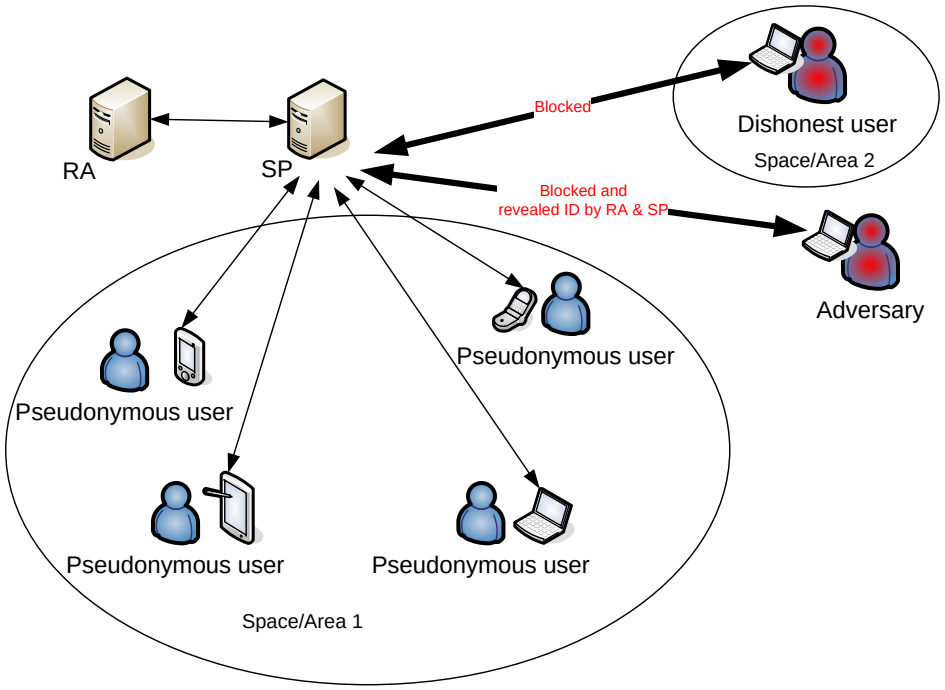


Fig. 9.1: Basic Scenario of Privacy-preserving Geosocial Applications.

by leaked position data. The situation can be changed by employing pseudonymous authentication schemes. These schemes provide fundamental security requirements like data integrity and authenticity as well as user privacy (by hiding identity information) and the revocation of malicious users.

9.1.1 Basic Scenario

In Figure 9.1, the basic scenario of privacy-preserving geosocial applications is depicted. A service provider usually provides a geosocial service, e.g. Foursquare or Yelp. Users can upload and share data such as their own reviews, photos or notifications without disclosing their identity. On the other hand, the authenticity of the data sent must be verified by the service provider or the user who obtains these data. Further, the users usually download the requested data about their location (space/area). These messages sent by users must be protected against disclosing users' identity to another party or eavesdroppers. Even the service provider is not able to recognize during downloading who the concrete data are for. The users in an area stay anonymous under their pseudonyms so they are pseudonymous. If a dishonest user creates bogus data that have a negative impact on the users in the geosocial service, then SP revokes their pseudonyms and blocks their next contributions to the geosocial service. In case a user repeatedly breaks the rules of SP, then

SP requests the registration authority to reveal his/her ID and revokes the adversary permanently. The problem of privacy protection and security must be addressed in geosocial services where users share their data which contain sensitive and private data.

One from main goals of this thesis is to develop and implement a secure and efficient cryptographic protocol providing privacy to users. By using the protocol, geosocial services become more trustworthy and secure for users and providers. Therefore, it is proposed a comprehensive privacy preserving framework that can be employed in privacy-friendly geosocial applications (see the scenario in Figure 9.1). The designed solution maintains users in anonymity unless they break the rules of the system and so the users are pseudonymous. Because some geosocial services require exact locations, the obfuscation of user location, for example by spatial cloaking, can compromise the utility of services. The proposed solution maintains the exact locations of pseudonymous users. In this chapter, it is proposed a novel framework that provides user pseudonymity and security designed for geosocial applications. The framework employs advanced cryptographic primitives considered secure nowadays. User privacy is maintained as well as data authenticity, integrity and confidentiality in both upload and download communication.

9.2 Privacy in Geosocial Services

Despite the current expansion of geosocial services on smartphones, there is no application that protects user privacy and is robust as regards unauthorized profiling. However, several theoretical proposals try to solve privacy in geosocial services. The importance of user anonymity and privacy in on-line electronic services is investigated in several works such as [96], [160], [92] and [171]. The work [148] surveys the basic features of existing geosocial services and the privacy issues involved in geosocial networks.

In this chapter, the features of popular geosocial services and user identity protection are outlined. Further, the authors consider that spatial and temporal cloaking is not appropriate for some geosocial services. On the other hand, they claim that encryption-based techniques could introduce additional system costs.

The work [98] investigates location privacy threats and countermeasures like using anonymity, spatial and temporal degradation. The techniques using the spatial and temporal degradation obfuscate the precise location of user and time data. The recent techniques of anonymity like changing pseudonyms by mix zones [16], dummy/fake positions [88], [95], anonymous attributes [77] or k-anonymity [159] trade-off the tractability of the privacy against the accuracy of location. Further,

the work [59] proposes to use different levels of granularity (the subset of RFC4119's civic location format). Users can blur the latitude and longitude coordinates by a parameter of a specified magnitude. Nevertheless, approaches using an obfuscation of the location are not suitable for the basic scenario, where the service provider demands a precise localization. Due to this fact the approach based on encryption-based techniques is more suitable to ensure privacy.

The paper [178] proposes a security mechanism including a key-exchange protocol and a distributed content access authorization scheme not relying on servers and trusted third parties. The authors use ECDH, MD5 and AES. First, the users must establish a shared AES key by means of ECDH with an interlock mechanism that protects against the MitM attack. This process takes 8 messages. Then, there is a content key to secure a confidential group communication. The privacy of users is protected by encryption messages and shared only among friends in the system. This protection does not provide user privacy but only the privacy of their data. The work [142] proposes a similar protection of user privacy in location-based social applications. The proposal addresses the privacy protection of users (user groups) by exchanging secret keys for symmetric cryptography. Another approach presented in [126] is also based on symmetric cryptography and hash functions. The paper focuses on the location privacy problem in proximity services. These concepts are efficient but applicable only in some scenarios in geosocial applications where group members must contact each other and establish a common secret key.

The paper [43] proposes a privacy preserving scheme of geolocation badge services (e.g. Foursquare or Yelp). Blind signatures, an anonymizer (e.g. Tor), Quick Response Codes (QR codes or 2D barcodes) and an anonymous authentication scheme based on a zero knowledge method are used to verify the claimed location and preserve user privacy at the same time. Nevertheless, the proposed scheme is closely related to making check-ins, putting forward recommendations and collecting prize badges. The locations and venues of honest users are anonymous. The scheme protects the service provider against badges being counterfeited by malicious users. Other privacy-preserving concepts based on cryptographic schemes usually employ many bilinear pairing operations, such as in [103]. In practice, however, these schemes are slow on some restricted devices like mobiles, since the bilinear pairing operation is currently computationally very expensive (in the best case 7.5x more expensive than exponentiation [151]) Due to this fact, it is important to optimize schemes and use a minimum of pairing operations. The work [55] deals with the implementation of a secure two-party computation for smartphones with an application to a privacy-preserving interest-cast. The paper outlines a feasible framework for smartphone environment and the implemented protocol called MobileFairPlay, derived from the FairPlay framework [124]. The performance of the implemented

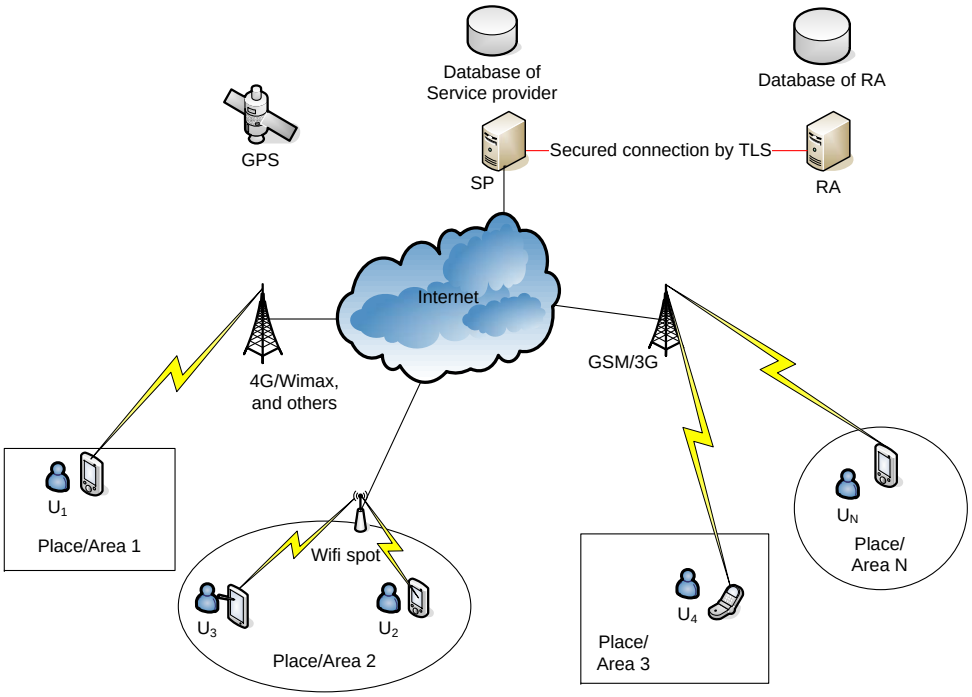


Fig. 9.2: System Model.

MobileFairPlay scheme takes, however, as much as about 5 sec.

In this chapter, it is proposed a privacy-preserving security framework which is convenient to implement on smartphones. Moreover, the solution does not use the privacy approach with location obfuscation.

9.3 Privacy-Preserving Framework for Geosocial Services

In this part, the proposed privacy-preserving framework for geosocial services is presented. The communication pattern of the solution, requirements and cryptographic techniques used in the proposed framework are outlined. The framework focuses on the practical and secure user registration and logging into a privacy preserving geosocial service. The framework maintains security, user privacy and efficiency during the upload and download of messages between users and a service provider.

9.3.1 System Model of Framework

The system model of the designed solution, which is depicted in Fig. 9.2, consists of a Registration Authority (RA), a Service Provider (SP) and a User (U).

- **RA** issues certified registration pseudonyms and can reveal the real identity of a user in the permanent revocation phase. RA is securely connected with a service provider and manages the registration of all users as a trusted third party in the framework.
- **SP** manages geosocial applications and services in the framework. It is assumed that SP is partly honest and is securely connected with the RA. SP generates group secret keys and sends them to members in the logging phase. SP is able to trace and open bogus messages. On the other hand, SP cannot reveal the real user identity and cannot create the profiles of user behavior. Nevertheless, SP issues the reputation of pseudonymous users and specifies fees for users who break rules in geosocial services.
- **U** with the certified registration pseudonym can upload and download the messages of geosocial applications. Before sending the messages, U has to register with RA and log in to geosocial services that are managed by SP. Further, U can report bogus messages to SP and help SP to track adversaries in the environment of the geosocial service.

9.3.2 Security Requirements of Framework

The proposed framework meets these security requirements:

- *Authenticity.* In upload connection, the message signatures must be created only by users who hold a valid and fresh group member secret key pair. In download connection, the signature scheme provides that messages are signed by the service provider who holds a private key.
- *Non-repudiation* Users or a service provider cannot deny that they created the signed messages.
- *Message Integrity.* Messages cannot be modified once they have been sent.
- *Message confidentiality.* The framework offers message confidentiality as an option in the encrypted upload connection and in the encrypted requested download connection. The user can download confidential data from a service provider and upload confidential data to a service provider. An attacker cannot reveal the meaning of data without proper private keys. Message confidentiality is ensured by probabilistic encryption/decryption.
- *Pseudonymity (user privacy).* An honest user U can use the registration pseudonym signed by RA to obtain group signature parameters and keys from a service provider. Then, U signs every upload message or request on behalf of the group members. Nobody is able to reveal user identity from the messages besides the RA and SP who collaborate.
- *Message unlinkability.* All messages, including identical messages, sent by the

user to SP cannot be linked by other users and internal or external adversaries. Message granularity protects against creating the profiles of users' behaviors. The pseudonyms of all group signatures are unique and random.

- *Revocation.* The solution protects user's privacy. Nevertheless, every malicious user is revealed due to the collaboration of SP and RA. If some user breaks the rules, his/her messages are opened by SP. Then, depending on the contravention, their pseudonym can be sent to RA, who is able to extract the user's identity. The next time an attacker tries to request a new pseudonym with a fresh time stamp, RA checks if their identity is in the list of permanently revoked users.

9.3.3 Cryptographic Components Used in Framework

The framework uses three main cryptographic components:

- A short **group signature scheme** to ensure privacy, authenticity, message integrity and non-repudiation and unlinkability. A modified short group signature based on the BBS04 scheme [20] is used. This pairing-based scheme ensures user pseudonymity in the upload and download connections of geosocial applications. The scheme is based on the Decision Linear and q-SDH problems, which are described in [20].
- A **digital signature scheme** to ensure authenticity, message integrity and non-repudiation. The framework implements the ECDSA signature scheme [93] with the public/private, i.e. verification/signature, keys of RA, SP and U. ECDSA is employed in the logging and registration of users and in download connection.
- A probabilistic **encryption/decryption scheme** to ensure message confidentiality, message integrity and unlinkability. The framework uses the probabilistic ElGamal encryption/decryption [164] during the logging and registration of users and, additionally, in the encrypted download and upload connection.

9.4 Framework Phases

In this section, the framework phases are outlined. Every phase can contain one or more processes that are defined in this section. Notation used in the framework is summarized in Table 9.1. The framework consists of ten main phases:

- *Initialization.* RA, SP and U set up cryptographic parameters and keys.
- *Registration.* RA $\Longleftrightarrow$ U: RA registers a user with installed geosocial applications.
- *Logging.* U $\Longleftrightarrow$ SP: Users log in to a geosocial service issued by SP.

Tab. 9.1: Notation Used in Framework.

$\Longleftrightarrow$	an encrypted bidirectional communication	$\longleftrightarrow$	a bidirectional communication
$\Longrightarrow$	an encrypted one-way communication	$\longrightarrow$	a one-way communication
$\hookleftarrow$	open user signature	$\|\|$	concatenation
Name (*input*) $\rightarrow$ *output*	an algorithm definition	Act_i	actual data about areas
A_i	a part of a member secret key	α	a random element $\in Z_p^*$
β	a random element $\in Z_p^*$	χ	an element of a group manager secret key $\in Z_p^*$
c	a hash value in the group signature / self-challenge $c \overset{R}{\leftarrow} Z_q$	cer_{U_i}	users' certificate signed by RA
δ	a commitment value in a signature	$e()$	a pairing operation
$enc_{pk_{RA}}$	an ElGamal encryption by RA	$enc_{pk_{U_i}}$	an ElGamal encryption by U
ϵ	a signature by SP	γ	a secret element $\in Z_p^*$
g_1	a generator of G_1	g_2	a generator of G_2
G_1	a multiplicative cyclic group of a prime order p	G_2	a multiplicative cyclic group of a prime order p
$gmsk$	a group manager secret key	gpk	a group public key
gsk_{U_i}	a group member secret key	h	a chosen element $\in G_1^*$
H	a hash function	ch	a challenge $c \overset{R}{\leftarrow} Z_q$
ID_{area}	an area ID	ID_{U_i}	a user ID
M	a message	μ	a commitment value in a signature
n	a number of signatures	ν	the element of a group manager secret key $\in Z_p^*$
π_{U_i}	the user certificate issued by RA	p_i	a temporary result of the pairing
pk_{SP}	an ElGamal public key of SP	pk_{RA}	an ElGamal private key of RA
pk_{U_i}	an ElGamal private key of a user	PRL	a Permanent Revocation List
Q	a query	r	a random reference number
$r_\alpha, r_\beta, r_x, r_\delta, r_\mu$	a random elements $\in Z_p^*$	Res	a response message
R_i	a commitment value in a signature	s	elements in signature$\in Z_q$
sig_{SP}	an ECDSA private key of SP	sig_{RA}	an ECDSA private key of RA
sig_{U_i}	an ECDSA private key of a user	sk_{SP}	an ElGamal private key of SP
sk_{RA}	an ElGamal private key of RA	sk_{U_i}	an ElGamal private key of a user
σ	the product of a group signature	T_i	pseudonyms in a signature
TRL	a Temporary Revocation List	t_l	a time stamp
θ	random elements $\in Z_p$	u	an element of a group public key
v	an element of a group public key	ver_{RA}	an ECDSA public key of RA
ver_{SP}	an ECDSA public key of SP	ver_{U_i}	an ECDSA public key of a user
w	an element of a group public key	x_i	an element of a group member secret key
Z_p	the (set of) p-adic integers	Z_q	the (set of) q-adic integers

- *Upload.* U $\longrightarrow$ SP: Users send signed messages to SP and SP verifies the signed messages from the user.
- *Encrypted Upload.* U $\Longrightarrow$ SP: The Upload phase using the encryption and decryption of confidential messages from users.
- *Requested Download.* SP $\longleftrightarrow$ U: SP sends download data to users in response to their request.
- *Encrypted Requested Download.* SP $\Longleftrightarrow$ U: The Requested Download phase using the encryption and decryption of confidential messages from SP.
- *Multicast Download.* SP $\longrightarrow$ U: SP sends download data to the groups of users without any requests.
- *Temporary Revocation.* SP $\hookleftarrow$ U: SP revokes a user who breaks the rules of a geosocial service for a short time, without revealing his/her identity.
- *Permanent Revocation.* RA $\Longleftrightarrow$ SP $\hookleftarrow$ U: RA collaborating with SP revokes a user for a long time and reveals his/her identity.

The registration, logging and upload phases are based on the cryptographic engine presented in [114] and are modified for use in geosocial services.

9.4.1 Initialization

The initialization phase generates all cryptographic parameters including public and private keys.

- RA initializes the parameters $(G_1, G_2, g_1, g_2, \psi, e)$, generates an ECDSA key pair sig_{RA}/ver_{RA}, an ElGamal private key sk_{RA}, and a public key pk_{RA}, and releases the public keys and parameters.
- SP generates group signature keys, ElGamal private sk_{SP} and public pk_{SP} keys using for secure publishing the group user secret keys. To secure download communication, SP generates an ECDSA key pair sig_{SP}/ver_{SP} and releases the public ECDSA key ver_{SP}. To issue the group signature scheme, SP randomly selects $\chi, \nu \in Z_p^*, h \in G_1^*$. Then, SP sets $u, v \in G_1^*$ such that $u^\chi = v^\nu = h$ and computes $w = g_2^\gamma$ such that $\gamma \in Z_p^*$ is randomly selected. The group public key is $gpk = (g_1, g_2, u, v, w, h)$ and the group manager secret key is $gmsk = (\chi, \nu)$.
- Users create ECDSA key pairs sig_{U_i}/ver_{U_i} to secure the registration and logging phases.

9.4.2 Registration

In the registration phase, the i-th user U_i installs a geosocial application on his/her device and obtains the entry credential number (a serial number of installed appli-

cation signed by SP) from SP. For the first time, RA must verify the user's real identity and his/her entry credential number (e.g. the serial number) obtained from SP. Then, U_i gives the ECDSA public key to RA, which stores (ID_{U_i}, ver_{U_i}) with the entry credential number in the database. RA then signs and returns a certificate $cer_{U_i} = sig_{RA}(ID_{U_i}, ver_{U_i})$ to U_i.

After successfully issuing the signed certificate, U_i requests a valid registration pseudonym π_{U_i} from RA. It is assumed that U_i has $cer_{U_i}, pk_{RA}, ver_{RA}$ containing the user's certificate, the public ElGamal key and the ECDSA key of RA, respectively.

The `registration process` $\mathbf{Reg}(ID_{U_i}) \rightarrow \pi_{U_i}$ proceeds in these steps:

1. U_i self-generates an ElGamal key pair (sk_{U_i}/pk_{U_i}) and sends the encrypted request $enc_{pk_{RA}}(pk_{U_i}||ID_{U_i}||\ sig_{U_i}(pk_{U_i}))$ to RA using the public ElGamal key of RA.
2. RA decrypts the ciphertext containing the request. Then, RA checks if the ID_{U_i} is not revoked and included in the Permanent Revocation List (PRL) and checks the user's signature, which ensures user's authenticity. Finally, the pk_{U_i} in the certificate cer_{U_i} with a new ElGamal key pair are committed and a challenge $ch \xleftarrow{R} Z_q$ and a time stamp t_l are generated and the encrypted response $enc_{pk_{U_i}}(enc_{pk_{RA}}(ID_{U_i}||\ ver_{U_i}||ch)||t_l||\ sig_{RA}(t_l\ ||enc_{pk_{RA}}\ (ID_{U_i}||ver_{U_i}||ch)||pk_{U_i}))$ is sent back to U_i.
3. U_i checks RA's signature and composes the registration pseudonym $\pi_{U_i} = pk_{U_i}||enc_{pk_{RA}}(ID_{U_i}||\ ver_{U_i}||ch)||t_l||\ sig_{RA}(t_l||enc_{pk_{RA}}(ID_{U_i}||ver_{U_i}||ch)\ ||pk_{U_i})$ and stores it.

If the registration pseudonym π_{U_i} expires (t_l is older than is tolerable in the geosocial application), the user needs to refresh his/her pseudonym by the registration process $\mathbf{Reg}(ID_{U_i}) \rightarrow \pi_{U_i}$.

9.4.3 Logging

The user U_i, who is using the installed geosocial application issued by SP, is logging into the service by requesting the group public key and his/her group member secret key.

The `logging process` $\mathbf{Log}(\pi_{U_i}) \rightarrow gsk_{U_i}$ follows these steps:

1. U_i sends his/her registration pseudonym $\pi_{U_i} = pk_{U_i}||enc_{pk_{RA}}(ID_{U_i}||ver_{U_i}||\ ch)||t_l||\ sig_{RA}(t_l||enc_{pk_{RA}}(ID_{U_i}||ver_{U_i}||ch)||pk_{U_i})$, which is encrypted by the ElGamal scheme using the SP public key pk_{SP}, to SP.
2. SP decrypts π_{U_i} by its own private ElGamal key sk_{SP}, and verifies π_{U_i} by the public ECDSA key ver_{RA} and checks if $enc_{pk_{RA}}(ID_{U_i}||ver_{U_i}||ch)$ is not on the Temporary Revocation List (TRL). If π_{U_i} is valid, SP creates $gsk_{V_i} = (x_i, A_i)$, where $x_i = H(enc_{pk_{RA}}(ID_{U_i}||\ ver_{U_i}||ch)||t_l||\gamma), A_i = g_1^{\frac{1}{x_i+\gamma}}$ and $H()$ is a hash

function. Then, SP stores $(enc_{pk_{RA}}(ID_{U_i}||\ ver_{U_i}||ch), A_i, t_l)$ in the logging table and sends gsk_{U_i} encrypted by pk_{U_i} to U_i.

3. U_i decrypts the message by its own private ElGamal key sk_{U_i} and checks the x_i by the hash function.

SP also verifies the validity of the time stamp t_l. Depending on the geosocial service policy, the π_{U_i} with the time stamp t_l can be marked as expired after a certain period. If U_i wants to continue using the geosocial service, they need to register a new pseudonym with fresh t_l. This approach reduces the size of TRL. The ElGamal encryption/decryption is probabilistic, hence eavesdroppers cannot link two or more encrypted messages if U_i requests gsk_{U_i} for a second time, i.e. the next user logs in $\mathbf{Log}(\pi_{U_i}) \rightarrow gsk_{U_i}$.

9.4.4 Upload

The communication pattern of upload connection is depicted in Figure 9.3. The users upload their data to share them in geosocial services. The upload phase employs the short group signature scheme, namely the BBS04 scheme [20].

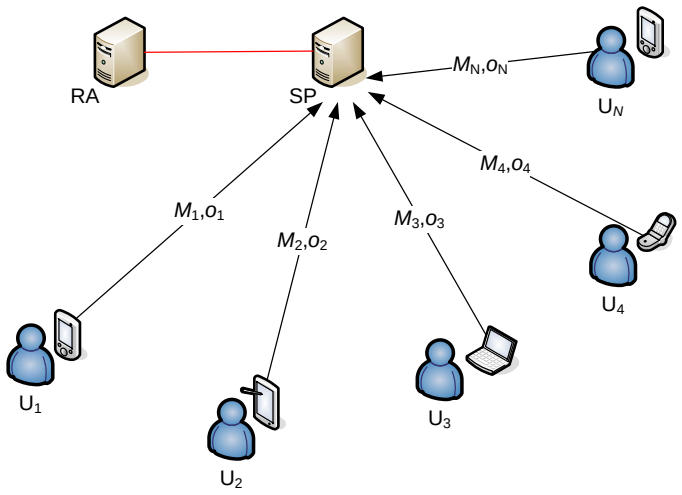


Fig. 9.3: Upload Connection.

Upload - Signing

Every user U_i who wants to send a new message to SP has to sign the message. Every U_i has a member secret key $gsk_{U_i} = (x_i, A_i)$ and a group public key $gpk = (g_1, g_2, h, u, v, w)$. U_i signs a message $M \in \{0,1\}^*$ and outputs the signature of knowledge $\sigma = (T_1, T_2, T_3, R_2, R_3, R_5, c, s_\alpha, s_\beta, s_x, s_\delta, s_\mu)$.

The `Upload Signing process` $\mathbf{UpSig}(M, gsk_{U_i}, gpk) \rightarrow \sigma$ consists of the next steps:

1. U_i generates random elements $\alpha, \beta, r_\alpha, r_\beta, r_x, r_\delta, r_\mu \in Z_p^*$,
2. computes

$$T_1 = u^\alpha, T_2 = v^\beta, T_3 = A_i h^{\alpha+\beta}, \tag{9.1}$$

$$\delta = \alpha x, \mu = \beta x, \tag{9.2}$$

$$p_1 = e(T_3, g_2), p_2 = e(h, w), p_3 = e(h, g_2). \tag{9.3}$$

3. stores $T_1, T_2, T_3, \delta, \mu, p_1, p_2, p_3$, and
4. computes

$$\begin{aligned} R_1 &= u^{r_\alpha}, \\ R_2 &= v^{r_\beta}, \\ R_3 &= p_1^{r_x} \cdot p_2^{-r_\alpha - r_\beta} \cdot p_3^{-r_\delta - r_\mu}, \\ R_4 &= T_1^{r_x} u^{-r_\delta}, \\ R_5 &= T_2^{r_x} v^{-r_\mu}, \end{aligned} \tag{9.4}$$

$$c = H(M, T_1, T_2, T_3, R_1, R_2, R_3, R_4, R_5), \tag{9.5}$$

$$\begin{aligned} s_\alpha = r_\alpha + c\alpha, s_\beta = r_\beta + c\beta, s_x = r_x + cx, \\ s_\delta = r_\delta + c\delta, s_\mu = r_\mu + c\mu. \end{aligned} \tag{9.6}$$

5. U_i sends the message M with the signature $\sigma = (T_1, T_2, T_3, R_2, R_3, R_5, c, s_\alpha, s_\beta, s_x, s_\delta, s_\mu)$.

The elements T_1, T_2, T_3 computed in Equation 9.1 are pseudonyms in the signature σ. The signatures carrying identical messages contain dissimilar pseudonyms due to the regeneration of parameters. Then a certain content that is periodically sent by a certain user cannot be linkable. Moreover, U_i can precompute Equations 9.1, 9.2, 9.3 and 9.4 and save time for the processing of real-time data. This reduces all 3 bilinear operations to 0, 10 exponentiations to 0 and 14 multiplications to 5.

The protection against denial of service attacks can be achieved by employing a short linkability and categorized verification presented in the Chapter 8 and in the paper [114]. For a short time period, the signer uses the same pseudonyms T_1, T_2, T_3 and random values α and β in steps 1 and 2. Then, SP can use a temporary list of honest users, who are known by the same pseudonyms, sort out the potential valid signed messages into the first priority level of verification and potential corrupt signatures into lower priority levels of verification.

Upload - Verification

The Service Provider (SP) verifies the messages received from pseudonymous users. The solution uses a batch verification for a more efficient verification process. The batch verification, investigated in [66], verifies n messages in one batch. First, the j-th received message M_j is checked by an SP server if it contains a valid time stamp, real and consistent data of geosocial applications. The precise value of the time stamp, or the time window, depends on a concrete geosocial application, communication technology used, distance with specific latency, etc. If the batch verification is invalid, then the divide-and-conquer approach is used to identify the invalid signatures that are verified in the individual verification.

The `Batch Verification process` $\mathbf{BaVer}(M_j, gpk, \sigma_j) \rightarrow$ valid/invalid, where SP uses $gpk = (g_1, g_2, h, u, v, w)$ to verify messages $\sigma_j = (T_{j1}, T_{j2}, T_{j3}, R_{j2}, R_{j3}, R_{j5}, c_j, s_{j\alpha}, s_{j\beta}, s_{jx}, s_{j\delta}, s_{j\mu})$ for $j = 1, ..., n$, does:

1. SP restores
$$\overline{R}_{j1} = u^{s_{j\alpha}} T_{j1}^{-c}, \overline{R}_{j4} = u^{-s_{j\delta}} T_{j1}^{s_x}, \tag{9.7}$$
2. computes a new control hash c'_j from the parameters received:
$c'_j = H(M_j, T_{j1}, T_{j2}, T_{j3}, \overline{R}_{j1}, R_{j2}, R_{j3}, \overline{R}_{j4}, R_{j5})$,
3. checks if $c'_j = c_j$. If yes, then SP continues with the verification, otherwise the message with the signature is inconsistent and is refused.
4. SP randomly selects $\theta_1, \theta_2, ..., \theta_n \in Z_p$ with l_b bit, checks the batch if
$$\prod_{j=1}^{j=n} R_{j3}^{\theta_j} = e(\prod_{j=1}^{j=n} (T_{j3}^{s_{jx}} h^{-s_{j\delta} - s_{j\mu}} g_1^{-c_j})^{\theta_j}, g_2) \; e(\prod_{j=1}^{j=n} (T_{j3}^{c_j} h^{-s_{j\alpha} - s_{j\beta}})^{\theta_j}, w) \tag{9.8}$$
and whether
$$1_{G_1} = (R_{j5} R_{j2})^{-\theta_j} T_{j2}^{\theta_j s_{jx} - \theta_j c_j} v^{(s_{j\beta} - s_{j\mu})\theta_j}. \tag{9.9}$$
5. If Equations 9.8 and 9.9 hold, the signed message is valid .

In case the batch verification is valid, then all messages from the batch continue into the application process of a geosocial service. Some of the valid messages can be resent to other relevant users in a certain area. These phases are described in the following subsections: *Requested Download* and *Multicast Download*.

In case the batch verification fails, then the divide-and-conquer approach is used to identify the invalid signatures that can be discarded.

At the end of the divide-and-conquer approach, the final two messages are individually verified.

The `Individual verification process` $\mathbf{InVer}(M, gpk, \sigma) \rightarrow$ valid/invalid consists of these steps:

1. SP restores
$$\overline{R}_1 = u^{s_\alpha} T_1^{-c}, \overline{R}_4 = u^{-s_\delta} T_1^{s_x}, \tag{9.10}$$
2. computes a new control hash c' from the parameters received:
$c' = H(M, T_1, T_2, T_3, \overline{R}_1, R_2, R_3, \overline{R}_4, R_5)$.
3. checks if $c' = c$. If yes, then SP continues with the verification, otherwise the message is inconsistent and is refused.
4. SP checks if
$$R_3 = e(T_3, g_2)^{s_x} e(h, w)^{(-s_\alpha - s_\beta)} e(h, g_2)^{(-s_\delta - s_\mu)} (e(T_3, w) e(g_1, g_2)^{-1})^c \tag{9.11}$$
and
$$1_{G_1} = (R_5 R_2)^{-1} T_2^{s_x - cx} v^{(s_\beta - s_\mu)}. \tag{9.12}$$
5. The signed message is valid if Equations 9.11 and 9.12 hold.

Equations 9.8 and 9.11 indicate that the individual verification requires 5 pairing operations per one message without optimization techniques and the batch verification requires only 2 pairing operations per n messages. Due to this fact, the scheme uses the batch verification in preference to the individual verification.

9.4.5 Encrypted Upload

The confidential message which is uploaded from U_i to SP, is encrypted by the ElGamal scheme.

U_i signs a message $M \in \{0,1\}^*$ by the process $\mathbf{UpSig}(M, gsk_{U_i}, gpk) \rightarrow \sigma$ and then they use the pk_{SP} to encrypt the message to a ciphertext. SP sk_{SP} decrypts the ciphertext to a plain text (the original message) and then they verify the group signature σ.

9.4.6 Requested Download

Every user U_i who wants to download the data of a geosocial service from SP sends a query Q, which is a request, to SP. The query message contains the location of the user, a time stamp, his/her request, and so on. The requested download phase is depicted in Fig. 9.4.

The `Requested Download process` $\mathbf{ReqDown}(r, Q, gsk_{U_i}, gpk) \rightarrow Res, \epsilon$ consists of these steps:

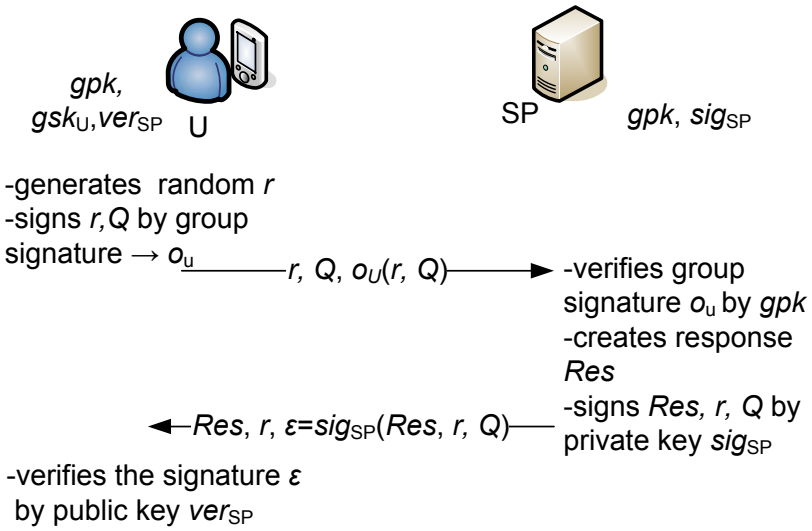


Fig. 9.4: Requested Download Connection.

1. U_i generates a random reference number r and sends a request that consists of the triplet r, Q, σ to SP. σ is the output of process $\mathbf{UpSig}(r, Q, gsk_{U_i}, gpk)$ where the random reference number r and query Q are signed by the user group secret key gsk_{U_i}.
2. SP verifies σ by gpk. Only a valid member of the group can request data successfully. Then, based on the query Q, SP compounds a response message Res containing the requested data. SP signs this message Res by his/her private ECDSA key sig_{SP}. SP sends Res and r in plain text and $\epsilon = sig_{SP}(Res, r, Q)$ back to the user.
3. U_i checks the signature ϵ by the SP public key ver_{SP}.

The group signature ensures authenticity, integrity and user privacy (pseudonymity and unlinkability). The authenticity and integrity of downloaded data are ensured by the ECDSA scheme.

9.4.7 Encrypted Requested Download

The encryption of requested download communication provides data confidentiality. If the requested data are confidential, the user encrypts his/her query Q and a random r by the public ElGamal key of SP pk_{SP}. The encrypted and signed (by gsk_{U_i}) query is sent to SP that can decrypt and verify the signature. Then SP encrypts the xor-ed response Res with the random r by the private ElGamal key sk_{SP} and sends this message to the user. The user decrypts the message by pk_{SP}, uses the stored random r and uses the xor function on the decrypted message to

obtain the response Res. The process of encrypted requested download phase is depicted in Fig. 9.5.

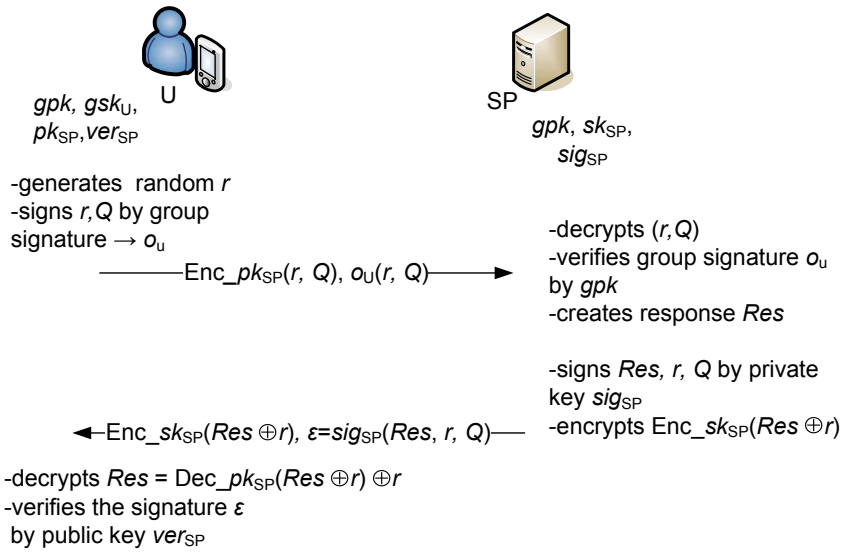


Fig. 9.5: Encrypted Download Connection.

9.4.8 Multicast Download

In multicast download connection, SP sends actual data Act about areas where active pseudonymous users are located. SP knows only the location of a user from his/her recent uploading or the query Q. These multicast messages are marked by an ID_{area} flag defining certain areas (e.g. streets, malls, districts, etc.). All messages that are sent from SP to users are signed by the SP private key sig_{SP}. Users check if the ID_{area} is correct and verify the downloaded messages by the SP public key ver_{SP}. The multicast download connection is depicted in Fig 9.6.

9.4.9 Temporary Revocation

Every message signed by a group member can be opened by SP using the group manager secret key $gmsk = (\chi, \nu)$. Bogus messages are messages with a correct signature but carrying a malicious content which breaks the policy of geosocial applications.

The `Temporary Revocation process` **TemRev**$(M, \sigma, gmsk) \rightarrow gsk_{U_i}, \pi_{U_i}$ is described in the following text:

SP extracts a part of the group member secret key $gsk_{U_i} \rightarrow A_i = T_3/(T_1^{\chi} \cdot T_2^{\nu})$ and searches the record $(enc_{pk_{RA}}(ID_{U_i}||ver_{U_i}||ch), A_i, t_l)$ in the database. SP identifies

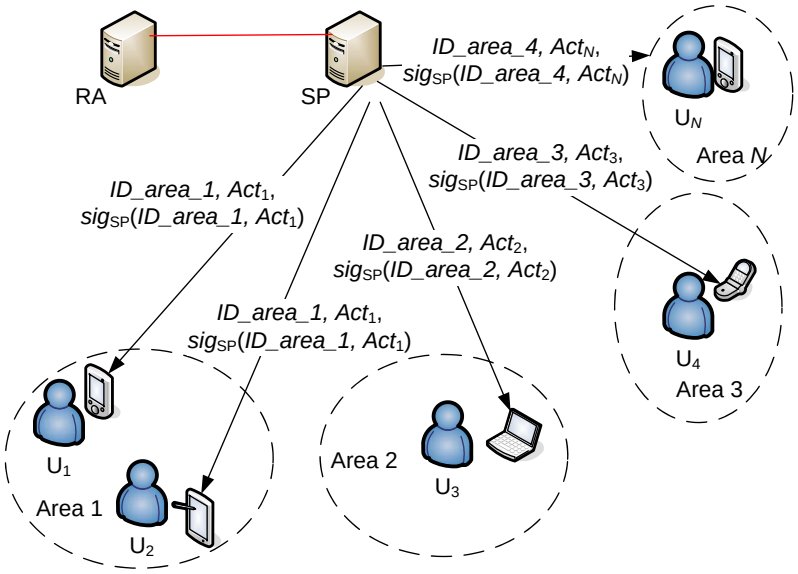


Fig. 9.6: Multicast Download Connection.

the registration pseudonym of the user π_{U_i} who sent the bogus message. SP saves this registration pseudonym in the Temporary Revocation List (TRL) until the lifetime of this registration pseudonym expires.

When the malicious user is logging to the system for the next time, SP does not issue a new group member secret key to the user so the user cannot sign and send more bogus messages. In serious cases, the registration pseudonym can be sent to RA for *Permanent Revocation*.

9.4.10 Permanent Revocation

In serious cases, such as breaking the rules repeatedly, a malicious user is revoked globally by the cooperation of SP and RA. SP is able to open a message and extract a group member secret key that is saved together with the user's registration pseudonym.

The `Permanent Revocation process` $\mathbf{PerRev}(\pi_{U_i}) \rightarrow ID_{U_i}$ is described in the following text:

SP sends the registration pseudonym of a user who seriously breaks the rules to RA. RA extracts ID_{U_i} by decrypting the registration pseudonym $enc_{pk_{RA}}(ID_{U_i} ||ver_{U_i}||ch), t_l)$ and adds ID_{U_i} to the Permanent Revocation List (PRL) so the malicious group member cannot update his/her pseudonym in the next registration phase. Moreover, RA can send ID_{U_i} and the serial number to SP and/or another government office (police) that can penalize the malicious user.

9.5 Security Analysis of Framework

This section presents possible attacks and a security analysis of the proposed framework according to the requirements given in 9.3.2. It is assumed that an attacker can control user devices and can also eavesdrop on, capture, modify and retransmit messages but his/her computational power does not permit the attacker to break current computationally secure cryptography schemes. Attackers can be external or also internal adversaries and can have access to SP database. As potential internal adversaries, SP or a user can be considered only. The Registration Authority (RA) is the trusted third authority that can be controlled by some governmental organization. Therefore, this entity is fully trusted. Regarding the Service Provider (SP), this entity is assumed to be managed by a private company that provides geosocial services. Then, it is expected that SP performs the phases in an honest way. It is expected that SP does not manipulate with messages, block them, etc. On the other hand, SP may try to retrieve users' identities to gather personal data and make users' profiles, which is a privacy threat. Hence, it is assumed that SP can try to break the privacy of users in the geosocial service.

9.5.1 Possible Attacks

Possible attacks can be divided into passive and active attacks. An attacker who performs a passive attack must have access to the communication. The main goal is to endanger user privacy by compromising the *message confidentiality*, *unlinkability* or both. The attacker performing passive attacks tries to track and link messages sent by a certain user, retrieve the real identity of a certain user or eavesdrop on messages transmitted between U_i and RA in the *Registration* phase, between U_i and SP in the *Logging* phase, *Encrypted Requested Download* and *Encrypted Upload* phase, and between RA and SP.

Active attacks tamper with valid messages, submit fake messages, etc. The usual purpose is to affect the normal execution of the proposed scheme to get some benefit. Active attacks compromise the message integrity, authenticity or both. The attacker performing active attacks usually tries to tamper with messages transmitted between U_i and RA in the *Registration* phase, between U_i and SP in the *Logging* phase, *Upload / Encrypted Upload / Requested Download / Encrypted Requested Download / Multicast Download* phase, and between *RA* and SP.

The attacker may then attempt to create fake but valid registration pseudonyms or certificates. Active attacks also include the generation of fake messages by unauthorized users, reusing the former messages to perform replay attacks or misusing the psedonymity without being traced and revoked.

9.5.2 Protection of the Framework against Possible Attacks

The behavior of the framework as regards the considered attacks is described and analyzed in the following text.

Protection of the Framework against Tracing of Messages Sent by a Certain User

All users sign messages by the modified short group signature BBS scheme [20]. Every generated group signature contains the group member's pseudonym T_1, T_2, T_3, which is a linear encryption of the member's secret key A_i and random elements α and β. If U_i generates new elements α and β, then the new generated signatures contain different pseudonyms T_1, T_2, T_3. Due to this fact, the group signature scheme used provides unlinkability of the former signatures with the new ones.

Behavior of the Framework as Regards Extracting the Identity of a Certain User

The framework contains only two phases where the user identity ID_{U_i} is a part of the process.

In the Registration phase, users show their ID_{U_i} only to a trusted RA, who issues a registration pseudonym π_{U_i} to the user. The pseudonym π_{U_i} containing the user ID_{U_i} is encrypted by the ElGamal public key pk_{RA}. Only RA with his/her ElGamal private key sk_{RA} can decrypt the pseudonym and reveal the user's ID_{U_i}.

In the Logging phase, users send their pseudonym to SP. The message is encrypted by the ElGamal private keys sk_{SP} but the pseudonym is decrypted by pk_{RA}. Hence, SP cannot retrieve the real user identity. SP can link all the request messages that contain the same π_{U_i}. On the other hand, if the user pseudonym π_{U_i} is updated with a certain frequency, then the pseudonym linkability is minimized. The frequency of the pseudonym updates affects the length of the Temporary Revocation List (TRL) and the unlinkability of users towards SP. If the frequency is high, then the size of TRL should be smaller and more user sessions are unlinkable but users have to register new pseudonyms more often, which burdens the communication overhead and the infrastructure. The pseudonyms should be released in advance to one user for several certain time periods and services. The optimal frequency then strongly depends on the concrete geosocial services, the number of users, the percentage of malicious users, etc.

If an attacker wants to retrieve the real user identity, they must decrypt the message and the pseudonym. This is unfeasible without the knowledge of the ElGamal private keys of RA and SP.

Protection of the Framework against Eavesdropping on the Messages

In the Registration phase, U_i sends a request $(enc_{pk_{RA}}(pk_{U_i}||ID_{U_i}||sig_{U_i}(pk_{U_i})))$ in order to get a new pseudonym and RA sends back a response $(enc_{pk_{U_i}}\ (enc_{pk_{RA}}(ID_{U_i}||\ ver_{U_i}||ch)||t_l||\ sig_{RA}(t_l||enc_{pk_{TA}}(ID_{U_i}||\ ver_{U_i}||ch)||pk_{U_i})))$. The response and request messages are encrypted using the ElGamal scheme, which is currently considered to be secure [164]. Because decryption requires the knowledge of the secret keys sk_{U_i} and sk_{RA}, the attacker cannot decrypt the cipher text of messages to get the transmitted data. The secret keys are only known by the honest user and the trusted registration authority.

In the Logging phase, the attacker cannot get the data transmitted between U_i and SP because the messages are also encrypted using the ElGamal scheme. The secret keys which are necessary to decrypt the sensitive information are sk_{U_i} and sk_{SP}. These keys are only known by the honest user and the service provider. It is assumed that SP behaves honestly.

In the Encrypted Requested Download and Encrypted Upload phases, the sensitive information is sent between U_i and SP. The attacker is not able to disclose this information from the encrypted messages because the messages from users to SP are encrypted using the ElGamal scheme. The attacker will be unable to decrypt them and get the transmitted data because decryption requires the knowledge of the secret key sk_{SP}. In the Encrypted Requested download phase, the messages from SP to users are encrypted by the secret key sk_{SP} and xor-ed by a random number r. Having the public key pk_{SP}, the attacker can perform the ElGamal decryption but without the knowledge of r they cannot get the data transmitted from SP to a certain user. Only the user knows a certain r and can get the data.

It is assumed that the connection between RA and SP is always secured using TLS, which uses asymmetric and symmetric cryptosystems to establish security properties like message authenticity, integrity and confidentiality.

Protection of the Framework against Tampering with Messages

In the Registration phase, message integrity and authenticity are ensured by the ECDSA signature scheme. The request messages contains the user public key pk_{U_i}, which is signed with the ECDSA signature key sig_{U_i}. Assuming that the ECDSA signature scheme and the hash function SHA-1 used are secure, then the ECDSA verification process detects if the request message has been modified.

In the Logging phase, message integrity and authenticity are ensured in a similar way. U_i sends its pseudonym, which RA signs using the ECDSA signature scheme. SP sends U_i its group member secret key $gsk_{U_i} = (x_i, A_i)$ encrypted by the user's ElGamal public key. The user can check if gsk_{U_i} is computed from his/her certificate

by the SHA-1 hash function. The integrity and authenticity of the group member secret key are provided by the hash function SHA-1 and ElGamal scheme.

In the Upload phase, the messages are signed and verified employing the modified short group signature BBS scheme [20]. This approach ensures message authenticity and integrity of the messages.

In the Requested Download phase, the messages transmitted between SP and U_i ensure integrity and authenticity by the ECDSA signature scheme (from SP to users) and by the modified short group signature BBS scheme (from users to SP).

In the Multicast Download phase, all downloaded messages transmitted between SP and U_i also ensure integrity and authenticity by the ECDSA signature scheme.

It is assumed that the connection between RA and SP, which uses TLS, is secured against tampering with messages.

Creating a Fake but Valid Pseudonym or User Message

To create a valid pseudonym π_{U_i}, the attacker needs the ECDSA private key sig_{RA}, which is only known by the RA. Due to this fact the attacker is not able to launch this attack. Moreover, if a fake pseudonym π_{U_i} is sent to a user, then they use the public ECDSA key ver_{RA} of RA to verify its validity. The user does not accept any fake pseudonyms.

The employed short group signature scheme ensures message authenticity, integrity and unlinkability of every message in the phases that use signing and verification processes. The proposed framework employs a modified short group signature BBS scheme [20] and inherits all its security features for the concrete phases and processes. Thus, only the service provider and the valid group member U_i are able to sign a message on behalf the group.

The attacker must recompute the hash c and some signature parts if they try to modify a certain message. It is assumed that the attacker cannot use the valid key $gsk_{U_i} = (A_i, x_i)$. Then, if the hash function used is secure and the Discrete Logarithm problem holds, computing the signature parts ($s_{j\alpha}$, $s_{j\beta}$, s_{jx}, $s_{j\delta}$, $s_{j\mu}$) without knowing x_i is considered unfeasible.

Protection of the Framework against Replay Attacks

In the designed framework, all messages contain a time stamp with the current time and date. The time stamp of each received message is checked before the verification of the group signatures or ECDSA signatures. Assuming that the attacker without a valid $gsk_{U_i} = (A_i, x_i)$ tries to reuse former messages with valid signatures, then they must also refresh the time stamp and recompute the hash c_j and the signature

$(s_{j\alpha}, s_{j\beta}, s_{jx}, s_{j\delta}, s_{j\mu})$. The computation of valid values for $s_{jx}, s_{j\delta}$ and $s_{j\mu}$ without knowing x_i is unfeasible under the Discrete Logarithm problem.

Behavior of the Framework Relative to Misbehavior of Pseudonymous Users

Every honest user is pseudonymous and unlinkable relative to other users and SP. If a user breaks the rules of geosocial services, e.g. sends malicious content, uses vulgarisms, etc., then the SP using its own group manager secret key $gmsk$ opens the message with correct signature and extracts part of the member secret group key $gsk_{U_i} \rightarrow A_i = T_3/(T_1^{\chi} \cdot T_2^{\nu})$. Then SP searches the record $(enc_{pk_{RA}}(ID_{U_i}||ver_{U_i}||c), A_i, t_l)$ in the database and decides if the user will be revoked temporarily or permanently. If the user who is temporarily revoked tries to log to a geosocial service, then SP can block their access and does not provide them with a new valid gsk_{U_i}. Moreover, that user does not receive any downloaded data and cannot request data from SP.

If the user breaks the rules periodically or behaves maliciously then SP can send his/her pseudonym to RA, who is able to retrieve the real user identity by ElGamal decryption and find in the database his/her serial number of the implemented geoservice application. Based on the serial number and the real identity of the user, SP or a government office can penalize the user.

9.6 Evaluation and Results of Framework

In this section, the cryptographic components used in the framework are evaluated. Further, an experimental results of employed cryptographic components and the framework phases are outlined.

9.6.1 Evaluation of Framework

The modified privacy preserving group signature scheme BBS04 [20] is employed in the Upload, Encrypted Upload, Requested Download, and Encrypted Requested Download phases. Section 6.1.1 shows that modular arithmetic operations like addition and subtraction can be computed more efficiently than multiplication T_m and exponentiation T_e or bilinear pairing T_p. Due to this fact, the fast operations can be omitted in this performance evaluation. The used group signature scheme achieves a more efficient batch verification ($2\ T_p + 11n\ T_e$) and individual verification ($5\ T_p + 10\ T_e$) than the pure BBS04 scheme [20] and Ferrara et al. [66] scheme due to the fewer exponentiation operations, likewise the scheme in Protocol 2. Further, the signing phase with precomputed parameters in the used scheme is more efficient

GS scheme:	Designed scheme (based on BBS04 [20])	BBS04 [20]	Ferrara et al. [66]
Batch:	yes	no	yes
Length of signature:	$5G_1, G_T, 5Z_p$ (2380 bits)	$3G_1, 6Z_p$ (1500 bits)	$3G_1, G_T, 6Z_p$ (2032 bits)
Performance of batch verification			
Pairings	**2**	5n	2
Exponentiation	**11n**	12n	13n
Multiplication	11n+1	**8n**	10n+1
Performance of individual verification			
Pairings	5	5	5
Exponentiation	**10**	12	12
Multiplication	9	8	8
Performance of normal mode signing / precomputed mode signing			
Pairings	3 / **0**	3	3
Exponentiation	12 / **0**	12	12
Multiplication	12 / **5**	12	12

Tab. 9.2: Evaluation of Group Signature Signing and Verification

than the signing of pure BBS04 scheme [20] or Ferrara et al. [66] scheme, see Table 9.2. The signing phase of the framework is more efficient than signing in Protocol 1 and similarly efficient as signing in Protocol 2.

The Requested Download, Multicast Download, Logging and Registration phases employ the ECDSA signature scheme. ECDSA is more efficient than group signatures due to the few inexpensive modular operations (modular multiplicative inverses, multiplications, additions) in signing and verification. On the other hand, the ECDSA signature scheme is not designed for privacy-preserving schemes. For every single ongoing message, users must create new triplets that contain a signature, a public key and a certificate to achieve the unlinkability of those messages. Therefore, this improper approach generates a lot of public keys and certificates. Due to this fact, the system entities must deal with huge amounts of keys and certificates in the revocation and issuing phases. For these reasons, the solution employs the ECDSA signature schemes only on non-privacy communication such as the registration phase or on messages that are signed by and downloaded from a service provider. The service provider can take advantage of the efficiency of the ECDSA signature scheme and sends the signed messages to users.

GS scheme:	Designed scheme	BBS04 [20]
Number of messages	Time [ms]	
Verification		
1	108	232
10	290	2297
20	977	4548
50	2301	11205
100	4681	22284
200	9251	44644
500	23294	111757
Signing		
1	157	226
5	498	861
10	786	1686
20	1326	3237
50	3106	7978

Tab. 9.3: Performance of Group Signature Schemes.

9.6.2 Experimental Implementation and Results

The cryptographic primitives have been implemented in JAVA. The experimental implementation is formed by three main cryptographic components. The first component is the ECDSA signature scheme provided by JAVA (JDK 7). Developers can also use the third party library such as the Bouncy castle Library[1], which offers more options and utilities. The ECDSA scheme implemented uses a 256-bit key size and employs the 256-bit SHA-1 hash function.

The second cryptographic component is a group signature scheme that uses the Java Pairing Based Cryptography (jPBC) Library[2]. The implementation employs the MNT curves type D with the embedding degree $k = 6$, the 171-bit order of curves and the pre-generated parameters d840347-175-161.param.

The registration and logging phases use the ElGamal encryption, which is the third cryptographic component in the experimental implementation. All ElGamal keys can be created by class org.bouncycastle.jce.provider.JDKKeyFactory. The 1024-bit ElGamal encryption is used. The framework is tested on a PC machine with Intel(R) Xeon(R) CPU X3440 @ 2.53GHz, 4 GB Ram, Windows 7 Professional.

Table 9.3 shows the results measured for the used group signature scheme (signing, verification), which is compared with pure BBS04 scheme [20]. The verification

[1](available on http://www.bouncycastle.org/resources.html)
[2](available on http://gas.dia.unisa.it/projects/jpbc/index.html)

ECDSA mode:	Signing	Verification	Signing	Verification
Size of messages:	500 B		8 kB	
Number of messages	Time [ms]			
1	2	3	3	3
10	18	30	24	32
50	82	143	93	152
100	163	288	184	294
500	802	1418	865	1482
1000	1594	2847	1638	2922

Tab. 9.4: Performance of ECDSA Signature Scheme.

Side:	User	SP/RA	User	SP/RA
Size of messages:	500 B		8 kB	
Framework phases:	Time [ms]			
Registration	61	15	-	-
Logging	6	11	-	-
Upload	157 (75*)	108 (50*)	157 (75*)	108 (50*)
Encrypted Upload	161 (79*)	110 (52*)	220 (138*)	139 (81*)
Requested Download	160 (78*)	110 (52*)	160 (78*)	111 (53*)
Encrypted Requested Download	166 (84*)	116 (58*)	254 (172*)	205 (147*)
Multicast Download	3	2	3	3

Tab. 9.5: Performance of Framework Phases per 1 Message / Request.

of the pure BBS04 scheme does not apply batch verification, which significantly affects the performance. The verification of the BBS04 scheme takes about 225 ms per one message on average. The batch verification of the scheme takes about 50 ms per one message on average. Moreover, it is assumed that a service provider who performs upload verification has at their disposal a more powerful machine so that the verification would be more efficient. Further, the upload signing in the framework takes about 75 ms per one message on average, which is a better result than for the BBS04 scheme, which performs the signing phase in 175 ms per one message on average. Assuming that users' smartphones usually are less powerful than the machine used, then it is expected a little bit longer signing. On the other hand, the user usually does not sign messages as frequently as a service provider does.

Table 9.4 shows the results measured for the ECDSA signature scheme. Two sizes of messages, 500 B and 8 kB, have been tested. The shorter length of message carries short check-in messages that consist of an ID venue, an event, a shout (e.g.

140 characters in Foursquare [56]), a user position, and so on. The 8 kB size messages should carry longer text messages, reviews, restaurant menus, programmes, and so on. The lengths of the output ECDSA signature are 70/71/72 B because they contain a random element. With the machine tested, the verification of 100 messages takes less than 300 ms and signing the 100 messages takes about 163 ms (for 0.5 kB messages), or 184 ms (for 8 kB messages). Assuming that service providers have at their disposal more powerful devices, they are able to secure more messages per the same time. The results listed in Table 9.4 indicate that the size of messages does not affect dramatically the performance of verification or signing. Due to the SHA-1 function, every message before signing or verifying is hashed to the 256 bit size.

The implemented 1024 bit ElGamal scheme is performed in 100 iterations and the average values measured are outlined. The ElGamal key initialization takes about 50 ms. The ElGamal encryption takes about 4 ms for 500 B messages or 63 ms for 8 kB messages. The ElGamal decryption takes about 2 ms for 500 B messages or 31 ms for 8 kB messages.

Table 9.5 shows the performance of the framework phases and steps. The total times of phases only include the performance of cryptographic components. The communication latency, data processing in application, etc. are not included. The registration and logging phases employ messages $\leq$ 500 B between a user and SP or RA, due to this fact the performance measurement with 8kB messages is omitted. The most frequently used phases such as the upload phase take about 125* (265) ms per 1 message on average, signing takes about 75* (157) ms and verification takes 50* (108) ms. The * marked results indicate average values that are computed via optimization techniques such as precomputation and batch verification. The upload with encryption takes about 131* (271) ms per one 500 B message on average and 219* (359) ms per one 8 kB message on average. The requested download phase takes 78* (160) ms on the user side and 52* (110) ms on the SP side. The framework offers more practical cryptography protection than the work [55], where the implemented privacy preserving framework takes about 5s to establish one secure two-party connection. Considering the total time of logging, upload and requested download phases, the framework takes about 220 (442) ms with (without) precomputation. If users need to transmit confidential data, then the Encrypted Upload and Requested Download phases offer message encryption. The performance of the ElGamal encryption used depends on the length of messages. For 8 kB messages the ElGamal encryption adds 63 ms and the ElGamal decryption adds 31 ms to the total time.

Furthermore, the framework phases have been implemented on computational restricted smartphones with the Android platform. The framework phases run on the user side are tested on a smartphone device, Samsung Galaxy S (I9000) with a 1

GHz Cortex-A8 processor, 512MB RAM and Android OS v2.3. The upload phase requires about 900 ms due to the precomputed signing mode of a group signature. The requested download phase takes 1150 ms in total without a communication delay. The verification of 1 ECDSA signature with 256 SHA-2 takes about 250 ms on the smartphone. Thus, the multicast download phase takes only 250 ms.

9.7 Summary of Chapter 9

The chapter introduces a cryptographic solution that preserves security, efficiency and user privacy in geosocial services. This privacy-preserving framework ensures message integrity and authenticity in the download and upload messages between users and a service provider that manages geosocial services and applications. Data confidentiality is also provided in encrypted upload and download phases to ensure more security and user privacy. Furthermore, the proposed solution provides pseudonymity and unlinkability for the users in front of other users and SP. User pseudonymity conceals user identity and unlinkability prevents creating profiles of users' behaviors. On the other hand, this protection can be revoked if the pseudonymous user misbehaves. The framework offers two levels of revocation. Temporary revocation is used if the user's misbehaviour is not very serious while permanent revocation is performed if a user behaves maliciously and/or breaks the rules periodically. Besides a trusted registration authority, no party is able to reveal the identity of a certain user. Geosocial services providing pseudonymity become more trustworthy among users that are concerned about their privacy.

The main cryptographic components and the framework phases such as upload (signing, batch and individual verification) and download are implemented and measured. The results show that the framework is efficient and is ready to be employed in geosocial services that connect thousands of users. Moreover, the user-side phases of the privacy preserving framework can be applied in geosocial applications where users use computational restricted devices like mobiles or smartphones.

10 DISCUSSION

In this chapter, the contribution of the proposed protocols from Chapters 7 and 8 is discussed. Besides the contribution of designed protocols, possible future extensions are outlined. Then, the proposed solution from Chapter 9 is discussed. Finally, the overall contribution and accomplishment of objectives of this thesis are summarized.

10.1 Discussion: Protocol 1

Protocol 1 (Chapter 7) provides standard group signature properties like authenticity, anonymity, data integrity, non-reputation, correctness and one public key. The scheme does not need the reinitialization of parameters and keys of members when a new user is added, revoked or epoch is ended. In contrary to schemes [131], [105], [48] and [28] where time intervals are employed, in the designed protocol, a Revocation List (RL) is reduced by the natural expiration of secret keys which is convenient for applications where the individual time of group membership expiration is needed. To actual best knowledge, only the scheme proposed by Chu et al. 2012 [54] uses time-bound secret keys to the natural expiration of these keys. Nevertheless, the proposed protocol is more efficient in a computational overhead than Chu et al. scheme [54] by using a different design and employing optimization techniques such as the batch verification. The protocol needs only 8 elements per a revocation token in contrary to 14 elements needed in [54]. According to the results in Section 9.6.2, the proposed protocol has better performance in the verification phase than the current VLR group signatures proposed in [131], [23], [28] and [54].

As future work, it would be worth including back unlikability into the proposed protocol by chopping the time-bound secret key. Further, the impact of the natural expiration on the revocation control for a large number of users can be investigated.

10.2 Discussion: Protocol 2

This section discusses Protocol 2 (Chapter 8) which deals with privacy-protection and security in VANET applications. The protocol focuses on providing security and privacy protection and tries to offer efficient signing and verification processes. In addition to those features, it is aimed at the protection against denial of services attacks, which is not usually covered in the literature.

In the V2V communication, the protocol provides efficient signing with the short-term linkability. The proposal uses the modified scheme of Wei et al. (WLZ scheme) [169]. Nevertheless, the designed protocol adds short-term linkability obtaining

a more efficient signing phase than in the WLZ scheme. Moreover, the WLZ scheme is focused on the V2V communication and does not describe the registration and join phases in detail. The short-term linkability is demanded for several applications [157] and can protect against Sybil and Denial of Service attacks. Due to this, the proposed protocol in this thesis can provide an efficient categorized batch verification with this short-term linkability. Generally, in group signatures, the batch verification of n messages is more efficient than individual verification, but the complexity of a batch computation with bogus messages increases from O(1) to O(ln n). In [66], the authors claim that if $\geq 15\%$ of the signatures are invalid, then a batch verification is not more efficient than an individual verification. The proposal modifies the WLZ scheme [169], where the batch verification costs only 2 pairings and $11n$ exponentiations. But the WLZ scheme and related solutions use an uncategorized batch verification which can cause less efficient verification if bogus messages appear during attacks like the Sybil attack, the Denial of Services (DoS) attack etc. However, the proposed protocol applies the categorized batch verification which sort potential honest messages to the first batch, and potential untrusted messages to the second or third batch with lower priorities, so the verification phase can be more efficient and strong against Sybil and DoS attacks.

In V2I communication, the designed protocol uses probabilistic cryptography for keeping the long-term unlinkability and privacy protection of drivers. The join or registration phase takes only two messages (request /response) and the protocol does not need tamper-proof devices. Moreover, the solution can avoid the inefficient linear growth of revocation list with the secret keys of members. Certified pseudonyms are valid to expiration date and, after the expiration date, certified pseudonyms are automatically revoked. Vehicles do not have to deal with a Revocation List (RL). Instead, the protocol uses only a Group Temporary Revocation List (GTRL) to deny malicious members accessing the group of VANET members.

As future work, the security of V2V communication should be done in more efficient way. At this stage, the designed protocol provides secure and private V2V communication for several participants, but in some cases the users can be more, and then the VANET application can be tardy in the message verification process.

10.3 Discussion: Framework

In the following text, the proposed framework from Chapter 9 is discussed.

The framework offers user privacy in geosocial applications. This privacy-preserving framework for geosocial applications provides that every honest user uses geosocial services without leaking his/her identity (ID) while downloading information and

uploading shared data. By using advanced cryptographic primitives, pseudonymous communication among users in geosocial applications is achieved. Moreover, the service provider is not able to reveal user ID and make user profiles. This is ensured by using a trusted third party. The cryptographic framework is practical, secure and efficient. In the proposed solution, the used cryptographic primitives are considered secure nowadays. The outlined security analysis vindicates the security of this solution. The experimental implementation proves the efficiency of the framework.

To make the framework more practical, two modes of user revocation are proposed. The modes of revocation depend on the infraction of a certain user. If a certain user breaks a soft rule of the service provider, they are temporarily revoked. A service provider issuing a geosocial service can block the user by a revocation list for a certain period. The size of the revocation list is periodically reduced by the expiration time of a certain user's credentials. The permanent revocation is applied to a user who behaves maliciously by breaking the rules periodically. His/her identity can be revealed, and he/she is revoked permanently by a trusted third authority.

10.4 Overall Contribution

In this section, the overall contribution of the thesis is summarized.

- **Security** The designed cryptographic protocols ensure the basic security properties such as authentication, integrity, confidentiality and non-repudiation. The designed protocols and the framework employ only secure cryptographic primitives and schemes with the secure lengths of parameters and keys. The proposed cryptographic solutions protect against well-know attacks and the security analyses are provided.
- **User privacy** The designed cryptographic solutions provide user privacy. Users communicate in a secure and privacy-friendly way. The cryptographic protocols protect users' private data such as (users' locations, personal informations, behavior manners). The authentication of users is designed as pseudonymous. The users are anonymous against the attackers and eavesdroppers, but in special cases their identity can be revealed by certain authorities in a communication system.
- **Efficiency** The solutions are designed to keep efficiency in a computational and communication overhead in huge heterogeneous networks with a lot of users having different end devices. These devices have different performance characteristics and different software specifications. The security properties are assigned to the certain types of data to decrease a burden of cryptographic protocols on systems and various end nodes. The phases of proposed protocols

are accelerated by optimization methods to get better performance mainly in scenarios where users process many messages in real-time.

- **Efficient key management** The key management is simplified. The proposed protocols based on group signature schemes use only one public key beside conventional digital signature such as RSA or ECDSA where every user has own public key that must be spread in a communication system. Further, the join phases which deal with member secret key distribution are designed to keep security and privacy.
- **Applicability** The proposed cryptographic protocol (Chapter 8) and the framework (Chapter 9) are designed for concrete communication systems used on heterogeneous networks, i.e. Vehicular Ad hoc Networks applications and geolocalization services.

10.5 Accomplishment of Thesis Objectives

The accomplishment of the thesis objectives presented in Chapter 3 is summarized as follows.

- Modern privacy-preserving cryptographic schemes such as group signature schemes and their security options on computationally restricted devices are analyzed and evaluated in Chapters 4, 5 and 6.
- The efficiency of cryptographic primitives and modular arithmetic operations such as multiplication, exponentiation, pairings, hash functions are measured on various devices (PCs, smartphones, smart cards) and the results are described in Section 6.1.
- In the thesis, two privacy-preserving cryptographic protocols based on pairing-based group signatures are proposed in Chapters 7 and 8. Both protocols focus on efficiency and user privacy. Protocol 1 deals with the efficient revocation. Protocol 2 aims at efficient signing and the verification of messages.
- Both proposed cryptographic protocols (1 and 2) are optimized by techniques such as precomputation and batch verification in basic phases such as signing the messages and the verification of signatures.
- The privacy-preserving cryptographic framework is introduced in Chapter 9 and is suitable for geosocial services run on heterogeneous networks. The framework uses a group signature, encryption and key establishment techniques to provide user privacy and data security.
- The security analyses of the proposed framework and Protocol 2 are provided. These analyses describe the protection of solutions against common active and passive attacks.

11 CONCLUSION

The main goal of this thesis is research in privacy-preserving protocols based on group signature schemes that can be employed in the heterogeneous networks using various types of end nodes. In the thesis, the two novel privacy-preserving protocols and one framework which are based on group signature schemes are proposed.

The first protocol aims at the revocation of users in group signature schemes. Protocol 1 provides an efficient user revocation which is based on the natural expiration of member secret keys. This protocol is designed to achieve the efficient verification phase. Due to this fact, the verifier is able to check more signatures of messages than verifiers who use related schemes. The drawback of this protocol is the signing phase which performs 2 computational expensive pairing operations.

The second protocol is designed for systems where user privacy, user revocation and data security are required. This group signatures-based protocol introduces a novel categorized batch verification that mitigates possible denial of service attacks in the vehicular ad hoc network applications. Protocol 2 ensures user privacy, long-term unlinkability, data authenticity, and integrity of messages. Moreover, users can compute expensive pairing operations that are needed for signing in advance due to the short-term linkability property. Thus, the signing phase can be more efficient, and the protocol can be implemented into delay-tolerant vehicular services that use restricted end nodes, e.g. smartphones or tablets. The security analysis proves that the protocol is secure.

The framework, as the third proposal in this thesis, provides a comprehensive secure, privacy-preserving solution for heterogeneous networks such as geosocial networks. Besides the message authenticity and integrity, user privacy, the framework offers the confidentiality of the messages. Due to the optimization techniques such as batch verification applied on the proposed framework, a computation overhead is reduced on the verifier's side. Moreover, the precomputation of the pairing operations in the signing phase makes this framework suitable for restricted devices such as smartphones, tablets and other devices used in geosocial services. The framework is proven to be secure by the security analysis provided.

The presented proposals and used group signature schemes in this thesis are verified by practical implementations. These implementations have been realized by the JAVA programming language and have been tested on the various types of the devices (e.g. personal computers, smartphones, tablets). The obtained results have been published at international conferences and in journals with an impact factor.

BIBLIOGRAPHY

[1] Abdalla, M.; An, J. H.; Bellare, M.; Namprempre, C.: From identification to signatures via the Fiat-Shamir transform: Minimizing assumptions for security and forward-security. In *Advances in Cryptology—EUROCRYPT 2002*, Springer, 2002, pp. 418–433.

[2] Acquisti, A.: The economics of personal data and the economics of privacy. In *texte de la conférence donnée en décembre*, 2010.

[3] Aloul, F.; Zahidi, S.; El-Hajj, W.: Two factor authentication using mobile phones. In *Computer Systems and Applications, 2009. AICCSA 2009. IEEE/ACS International Conference on*, may 2009, pp. 641 –644.

[4] Ardagna, C. A.; Jajodia, S.; Samarati, P.; Stavrou, A.: Providing users' anonymity in mobile hybrid networks. *ACM Transactions on Internet Technology (TOIT)*, vol. 12, no. 3, 2013: p. 7.

[5] Armknecht, F.; Sadeghi, A.-R.; Visconti, I.; Wachsmann, C.: On RFID Privacy with Mutual Authentication and Tag Corruption. In *Applied Cryptography and Network Security*, *Lecture Notes in Computer Science*, vol. 6123, ed. by J. Zhou; M. Yung, Springer Berlin / Heidelberg, 2010, ISBN 978-3-642-13707-5, pp. 493–510.

[6] Ateniese, G.; Camenisch, J.; Hohenberger, S.; de Medeiros, B.: Practical Group Signatures without Random Oracles. *IACR Cryptology ePrint Archive*, vol. 2005, 2005: p. 385.

[7] Ateniese, G.; Camenisch, J.; Joye, M.; Tsudik, G.: A practical and provably secure coalition-resistant group signature scheme. In *Advances in Cryptology—CRYPTO 2000*, Springer, 2000, pp. 255–270.

[8] Ateniese, G.; Song, D.; Tsudik, G.: Quasi-efficient revocation of group signatures. In *Financial Cryptography*, Springer, 2003, pp. 183–197.

[9] Bao, F.: An Efficient Verifiable Encryption Scheme for Encryption of Discrete Logarithms. In *Smart Card. Research and Applications*, *Lecture Notes in Computer Science*, vol. 1820, Springer Berlin / Heidelberg, 2000, pp. 213–220.

[10] Barreto, P. S. L. M.; Naehrig, M.; Politecnica, E.; Informationstechnik, L. F. T.: Pairing-Friendly Elliptic Curves of Prime Order. In *Proceedings of SAC 2005, volume 3897 of LNCS*, Springer-Verlag, 2005, pp. 319–331.

[11] Belenkiy, M.; Camenisch, J.; Chase, M.; Kohlweiss, M.; Lysyanskaya, A.; Shacham, H.: Randomizable Proofs and Delegatable Anonymous Credentials. In *Proceedings of the 29th Annual International Cryptology Conference on Advances in Cryptology*, Berlin, Heidelberg: Springer-Verlag, 2009, ISBN 978-3-642-03355-1, pp. 108–125.

[12] Belenkiy, M.; Chase, M.; Kohlweiss, M.; Lysyanskaya, A.: Non-Interactive Anonymous Credentials. 2007.

[13] Bellare, M.: Multi-signatures in the plain public-key model and a general forking lemma. In *In ACM CCS 06*, ACM Press, 2006, pp. 390–399.

[14] Bellare, M.; Garay, J.; Rabin, T.: Fast batch verification for modular exponentiation and digital signatures. In *Advances in Cryptology â€" EUROCRYPT'98*, *Lecture Notes in Computer Science*, vol. 1403, ed. by K. Nyberg, Springer Berlin, 1998, ISBN 978-3-540-64518-4, pp. 236–250.

[15] Bellare, M.; Micciancio, D.; Warinschi, B.: Foundations of Group Signatures: Formal Definitions, Simplified Requirements, and a Construction Based on General Assumptions. 2003.

[16] Beresford, A.; Stajano, F.: Mix zones: user privacy in location-aware services. In *Pervasive Computing and Communications Workshops, 2004. Proceedings of the Second IEEE Annual Conference on*, march 2004, pp. 127 – 131, doi:10.1109/PERCOMW.2004.1276918.

[17] Berthold, O.; Federrath, H.; Köpsell, S.: Web MIXes: A system for anonymous and unobservable Internet access. In *Designing Privacy Enhancing Technologies*, Springer, 2001, pp. 115–129.

[18] Bichsel, P.; Camenisch, J.; Groß, T.; Shoup, V.: Anonymous credentials on a standard java card. In *Proceedings of the 16th ACM conference on Computer and communications security*, CCS '09, New York, NY, USA: ACM, 2009, ISBN 978-1-60558-894-0, pp. 600–610.

[19] Bichsel, P.; Camenisch, J.; Neven, G.; Smart, N. P.; Warinschi, B.: Get shorty via group signatures without encryption. In *Security and Cryptography for Networks*, Springer, 2010, pp. 381–398.

[20] Boneh, D.; Boyen, X.; Shacham, H.: Short group signatures. In *Advances in Cryptology–CRYPTO 2004*, Springer, 2004, pp. 227–242.

[21] Boneh, D.; Gentry, C.; Lynn, B.; Shacham, H.: Aggregate and verifiably encrypted signatures from bilinear maps. In *Proceedings of the 22nd international conference on Theory and applications of cryptographic techniques*, EUROCRYPT'03, Springer-Verlag, 2003, ISBN 3-540-14039-5, pp. 416–432.

[22] Boneh, D.; Lynn, B.; Shacham, H.: Short Signatures from the Weil Pairing. In *Advances in Cryptology ASIACRYPT 2001*, *Lecture Notes in Computer Science*, vol. 2248, ed. by C. Boyd, Springer Berlin / Heidelberg, 2001, ISBN 978-3-540-42987-6, pp. 514–532.

[23] Boneh, D.; Shacham, H.: Group signatures with verifier-local revocation. In *Proceedings of the 11th ACM conference on Computer and communications security*, ACM, 2004, pp. 168–177.

[24] Boyen, X.: The uber-assumption family. In *Pairing-Based Cryptography–Pairing 2008*, Springer, 2008, pp. 39–56.

[25] Boyen, X.; Waters, B.: Compact group signatures without random oracles. In *Advances in Cryptology-EUROCRYPT 2006*, Springer, 2006, pp. 427–444.

[26] Boyen, X.; Waters, B.: Full-Domain Subgroup Hiding and Constant-Size Group Signatures. In *Public Key Cryptography PKC 2007*, *Lecture Notes in Computer Science*, vol. 4450, ed. by T. Okamoto; X. Wang, Springer Berlin / Heidelberg, 2007, ISBN 978-3-540-71676-1, pp. 1–15.

[27] Brands, S. A.: *Rethinking public key infraestructures and digital certificates: building in privacy*. The MIT Press, 2000.

[28] Bringer, J.; Patey, A.: Backward Unlinkability for a VLR Group Signature Scheme with Efficient Revocation Check. Tech. report, Cryptology ePrint Archive, Report 2011/376, 2011.

[29] Brogle, K.; Goldberg, S.; Reyzin, L.: Sequential aggregate signatures with lazy verification. In *Cryptology ePrint Archive: Listing for 2011*, 2011, pp. 1–30.

[30] Butun, I.; Wang, Y.; Lee, Y.-s.; Sankar, R.: Intrusion prevention with two–level user authentication in heterogeneous wireless sensor networks. *International Journal of Security and Networks*, vol. 7, no. 2, 2012: pp. 107–121.

[31] Camenisch, J.: The Camenisch-Lysyanskaya Private Credential System Explained. 2010.

[32] Camenisch, J.; Hohenberger, S.; Kohlweiss, M.; Lysyanskaya, A.; Meyerovich, M.: How to win the clonewars: efficient periodic n-times anonymous authentication. In *Proceedings of the 13th ACM conference on Computer and communications security*, CCS '06, New York, NY, USA: ACM, 2006, ISBN 1-59593-518-5, pp. 201–210.

[33] Camenisch, J.; Hohenberger, S.; Pedersen, M.: Batch Verification of Short Signatures. In *Advances in Cryptology - EUROCRYPT 2007*, *Lecture Notes in Computer Science*, vol. 4515, ed. by M. Naor, Springer Berlin / Heidelberg, 2007, ISBN 978-3-540-72539-8, pp. 246–263.

[34] Camenisch, J.; Kohlweiss, M.; Soriente, C.: An accumulator based on bilinear maps and efficient revocation for anonymous credentials. In *Public Key Cryptography–PKC 2009*, Springer, 2009, pp. 481–500.

[35] Camenisch, J.; Kohlweiss, M.; Soriente, C.: Solving revocation with efficient update of anonymous credentials. In *Security and Cryptography for Networks*, Springer, 2010, pp. 454–471.

[36] Camenisch, J.; Lysyanskaya, A.: An Efficient System for Non-transferable Anonymous Credentials with Optional Anonymity Revocation. In *Advances in Cryptology - EUROCRYPT 2001*, *Lecture Notes in Computer Science*, vol. 2045, 2001, pp. 93–118.

[37] Camenisch, J.; Lysyanskaya, A.: A signature scheme with efficient protocols. In *Proceedings of the 3rd international conference on Security in communication networks*, SCN'02, Berlin, Heidelberg: Springer-Verlag, 2003, ISBN 3-540-00420-3, pp. 268–289.

[38] Camenisch, J.; Lysyanskaya, A.: Signature Schemes and Anonymous Credentials from Bilinear Maps. In *Advances in Cryptology - CRYPTO 2004*, *Lecture Notes in Computer Science*, vol. 3152, 2004, pp. 1–6.

[39] Camenisch, J.; Stadler, M.: Efficient Group Signature Schemes for Large Groups (Extended Abstract). In *Proceedings of the 17th Annual International Cryptology Conference on Advances in Cryptology*, London, UK: Springer-Verlag, 1997, ISBN 3-540-63384-7, pp. 410–424.

[40] Camenisch, J.; Stadler, M.: Proof Systems for General Statements about Discrete Logarithms. Tech. report, IBM, 1997.

[41] Camenisch, J.; Van Herreweghen, E.: Design and implementation of the idemix anonymous credential system. In *Proceedings of the 9th ACM conference on Computer and communications security*, CCS '02, New York, NY, USA: ACM, 2002, ISBN 1-58113-612-9, pp. 21–30.

[42] Capkun, S.; Hubaux, J.-P.; Jakobsson, M.: Secure and privacy-preserving communication in hybrid ad hoc networks. In *F.-L. Wong and F. Stajano*, Citeseer, 2004.

[43] Carbunar, B.; Sion, R.; Potharaju, R.; Ehsan, M.: The Shy Mayor: Private Badges in GeoSocial Networks. In *Applied Cryptography and Network Security*, *Lecture Notes in Computer Science*, vol. 7341, ed. by F. Bao; P. Samarati; J. Zhou, Springer, 2012, ISBN 978-3-642-31283-0, pp. 436–454.

[44] Chacko, N. M.; Sam, S.; Leelipushpam, P. G. J.: A survey on various privacy and security features adopted in MANETs routing Protocol. In *Automation, Computing, Communication, Control and Compressed Sensing (iMac4s), 2013 International Multi-Conference on*, IEEE, 2013, pp. 508–513.

[45] Chatterjee, S.; Menezes, A.: On cryptographic protocols employing asymmetric pairings the role of ψ revisited. *Discrete Applied Mathematics*, vol. 159, no. 13, 2011: pp. 1311–1322.

[46] Chaum, D.: Blind Signatures for Untraceable Payments. In *CRYPTO*, 1982, pp. 199–203.

[47] Chaum, D.; Van Heyst, E.: Group signatures. In *Proceedings of the 10th annual international conference on Theory and application of cryptographic techniques*, EUROCRYPT'91, Berlin, Heidelberg: Springer-Verlag, 1991, ISBN 3-540-54620-0, pp. 257–265.

[48] Chen, L.; Li, J.: VLR group signatures with indisputable exculpability and efficient revocation. *International Journal of Information Privacy, Security and Integrity*, vol. 1, no. 2, 2012: pp. 129–159.

[49] Chen, L.; Page, D.; Smart, N.: On the Design and Implementation of an Efficient DAA Scheme. In *Smart Card Research and Advanced Application*, *Lecture Notes in Computer Science*, vol. 6035, ed. by D. Gollmann; J.-L. Lanet; J. Iguchi-Cartigny, 2010, pp. 223–237.

[50] Chen, Y.-M.; Wei, Y.-C.: SafeAnon: a safe location privacy scheme for vehicular networks. *Telecommunication Systems*: pp. 1–16, ISSN 1018-4864, 10.1007/s11235-010-9408-x.

[51] Chevallier-Mames, B.; Coron, J.-S.; McCullagh, N.; Naccache, D.; Scott, M.: Secure delegation of elliptic-curve pairing. In *Smart Card Research and Advanced Application*, Springer, 2010, pp. 24–35.

[52] Chim, T. W.; Yiu, S.-M.; Hui, L. C. K.; Li, V. O. K.: SPECS: Secure and privacy enhancing communications schemes for VANETs. *Ad Hoc Networks*, vol. 9, no. 2, 2011: pp. 189–203.

[53] Chow, R.; Jakobsson, M.; Masuoka, R.; Molina, J.; Niu, Y.; Shi, E.; Song, Z.: Authentication in the clouds: a framework and its application to mobile users. In *Proceedings of the 2010 ACM workshop on Cloud computing security workshop*, CCSW '10, New York, NY, USA: ACM, 2010, ISBN 978-1-4503-0089-6, pp. 1–6.

[54] Chu, C.; Liu, J.; Huang, X.; Zhou, J.: Verifier-Local Revocation Group Signatures with Time-Bound Keys. 2012.

[55] Costantino, G.; Martinelli, F.; Santi, P.; Amoruso, D.: An implementation of secure two-party computation for smartphones with application to privacy-preserving interest-cast. In *Privacy, Security and Trust (PST), 2012 Tenth Annual International Conference on*, IEEE, 2012, pp. 9–16.

[56] Cuddy, C.; Glassman, N.: Location-Based Services: Foursquare and Gowalla, Should Libraries Play? *Journal of Electronic Resources in Medical Libraries*, vol. 7, no. 4, 2010: pp. 336–343.

[57] Delerablée, C.; Pointcheval, D.: Dynamic fully anonymous short group signatures. In *Progress in Cryptology-VIETCRYPT 2006*, Springer, 2006, pp. 193–210.

[58] Dingledine, R.: Tor: anonymity online. 2012.

[59] Doty, N.; Mulligan, D. K.; Wilde, E.: Privacy Issues of the W3C Geolocation API. *CoRR*, vol. abs/1003.1775, 2010.

[60] Douceur, J.: The Sybil Attack. In *Peer-to-Peer Systems*, *Lecture Notes in Computer Science*, vol. 2429, ed. by P. Druschel; F. Kaashoek; A. Rowstron, Springer Berlin / Heidelberg, 2002, ISBN 978-3-540-44179-3, pp. 251–260.

[61] Dutta, R.; Barua, R.; Sarkar, P.: Pairing-Based Cryptographic Protocols: A Survey. *IACR Cryptology ePrint Archive*, vol. 2004, 2004: p. 64.

[62] El Aimani, L.; Sanders, O.: Efficient group signatures in the standard model. In *Information Security and Cryptology–ICISC 2012*, Springer, 2013, pp. 410–424.

[63] El Defrawy, K.; Tsudik, G.: ALARM: anonymous location-aided routing in suspicious MANETs. *Mobile Computing, IEEE Transactions on*, vol. 10, no. 9, 2011: pp. 1345–1358.

[64] El Defrawy, K.; Tsudik, G.: Privacy-preserving location-based on-demand routing in MANETs. *Selected Areas in Communications, IEEE Journal on*, vol. 29, no. 10, 2011: pp. 1926–1934.

[65] Fan, C.-I.; Hsu, R.-H.; Manulis, M.: Group signature with constant revocation costs for signers and verifiers. In *Cryptology and Network Security*, Springer, 2011, pp. 214–233.

[66] Ferrara, A. L.; Green, M.; Hohenberger, S.; Pedersen, M. O.: Practical Short Signature Batch Verification. In *Proceedings of the The Cryptographers' Track at the RSA Conference 2009 on Topics in Cryptology*, CT-RSA '09, Springer-Verlag, 2009, ISBN 978-3-642-00861-0, pp. 309–324.

[67] Fiat, A.: Batch RSA. *Journal of Cryptology*, vol. 10, 1997: pp. 75–88, ISSN 0933-2790.

[68] Fiat, A.; Shamir, A.: How to prove yourself: practical solutions to identification and signature problems. In *Advances in Cryptology—CRYPTO'86*, Springer, 1987, pp. 186–194.

[69] Fonseca, E.; Festag, A.; Baldessari, R.; Aguiar, R.: Support of Anonymity in VANETs - Putting Pseudonymity into Practice. In *Proceedings of IEEE Wireless Communications and Networking Conference (WCNC), Hong Kong*, March 2007.

[70] Freeman, D.; Scott, M.; Teske, E.: A taxonomy of pairing-friendly elliptic curves. 2006.

[71] Galbraith, S. D.; Paterson, K. G.; Smart, N. P.: Pairings for Cryptographers. *Discrete Appl. Math.*, vol. 156, no. 16, Sep. 2008: pp. 3113–3121, ISSN 0166-218X, doi:10.1016/j.dam.2007.12.010.

[72] Gerlach, M.; Festag, A.; Leinmuller, T.; Goldacker, G.; Harsch, C.: Security architecture for vehicular communication. In *The 5th International Workshop On Intelligent Transportation*, March 2007.

[73] Goldreich, O.: A short tutorial of zero-knowledge. 2010.

[74] Granger, R.; Page, D.; Smart, N. P.: High Security Pairing-Based Cryptography Revisited. In *In Algorithmic Number Theory Symposium ANTS VII, Springer-Verlag LNCS*, Springer, 2006, pp. 480–494.

[75] Grochla, K.; Stolarz, P.: Extending the TLS Protocol by EAP Handshake to Build a Security Architecture for Heterogenous Wireless Network. In *Computer Networks*, Springer, 2013, pp. 258–267.

[76] Groth, J.: Fully anonymous group signatures without random oracles. In *In proceedings of ASIACRYPT 06, LNCS series*, 2007.

[77] Hajny, J.; Malina, L.: Anonymous credentials with practical revocation. In *Satellite Telecommunications (ESTEL), 2012 IEEE First AESS European Conference on*, IEEE, 2012, pp. 1–6.

[78] Hajny, J.; Malina, L.: Unlinkable Attribute-Based Credentials with Practical Revocation on Smart-Cards. In *Proceedings of the Smart Card Research and Advanced Application Conference CARDIS 2012*, 2012, pp. 1–15.

[79] Hajny, J.; Malina, L.; Martinasek, Z.; Tethal, O.: Performance Evaluation of Primitives for Privacy-Enhancing Cryptography on Current Smart-Cards and Smart-Phones. In *Data Privacy Management and Autonomous Spontaneous Security*, Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2014, ISBN 978-3-642-54567-2, pp. 17–33, doi:10.1007/978-3-642-54568-9_2.

[80] Harn, L.: Batch verifying multiple DSA-type digital signatures. *Electronics Letters*, vol. 34, no. 9, apr 1998: pp. 870 –871, ISSN 0013-5194.

[81] He, D.; Bu, J.; Chan, S.; Chen, C.; Yin, M.: Privacy-preserving universal authentication protocol for wireless communications. *Wireless Communications, IEEE Transactions on*, vol. 10, no. 2, 2011: pp. 431–436.

[82] Horng, W.-B.; Lee, C.-P.; Peng, J.-W.: Privacy preservation in secure group communications for vehicular ad hoc networks. *Telecommunication Systems*: pp. 1–11, ISSN 1018-4864, 10.1007/s11235-010-9409-9.

[83] Hu, Y.; Laberteaux, K.: Strong VANET security on a budget. In *Proceedings of Workshop on Embedded Security in Cars (ESCAR)*, 2006.

[84] Huang, D.: Pseudonym-based cryptography for anonymous communications in mobile ad hoc networks. *International Journal of Security and Networks*, vol. 2, no. 3, 2007: pp. 272–283.

[85] Huang, Y.-M.; Hsieh, M.-Y.; Chao, H.-C.; Hung, S.-H.; Park, J. H.: Pervasive, secure access to a hierarchical sensor-based healthcare monitoring architecture in wireless heterogeneous networks. *Selected Areas in Communications, IEEE Journal on*, vol. 27, no. 4, 2009: pp. 400–411.

[86] Hussain, R.; Kim, S.; Oh, H.: Towards Privacy Aware Pseudonymless Strategy for Avoiding Profile Generation in VANET. In *Information Security Applications*, *LNCS*, vol. 5932, ed. by H. Youm; M. Yung, 2009, ISBN 978-3-642-10837-2, pp. 268–280.

[87] Hwang, J. Y.; Lee, S.; Chung, B.-H.; Cho, H. S.; Nyang, D.: Short group signatures with controllable linkability. In *Lightweight Security & Privacy: Devices, Protocols and Applications (LightSec), 2011 Workshop on*, IEEE, 2011, pp. 44–52.

[88] Jensen, C.; Lu, H.; Yiu, M.: Location privacy techniques in client-server architectures. *Privacy in Location-Based Applications*, 2009: pp. 31–58.

[89] Jo, H.; Paik, J.; Lee, D.: Efficient Privacy-Preserving Authentication in Wireless Mobile Networks. 2013.

[90] Joux, A.: The Weil and Tate Pairings as Building Blocks for Public Key Cryptosystems. In *Algorithmic Number Theory, 5th International Symposium, ANTS-V, Sydney, Australia, July 7-12, 2002, Proceedings*, *Lecture Notes in Computer Science*, vol. 2369, ed. by C. Fieker; D. R. Kohel, Springer, 2002, ISBN 3-540-43863-7, pp. 20–32.

[91] Juels, A.: RFID Security and Privacy: A Research Survey. *JOURNAL OF SELECTED AREAS IN COMMUNICATION (J-SAC)*, vol. 24, no. 2, 2006: pp. 381–395.

[92] Kalloniatis, C.; Kavakli, E.; Gritzalis, S.: Addressing privacy requirements in system design: the PriS method. *Requirements Engineering*, vol. 13, 2008: pp. 241–255, ISSN 0947-3602, doi:10.1007/s00766-008-0067-3.

[93] Karati, S.; Das, A.; Roychowdhury, D.; Bellur, B.; Bhattacharya, D.; Iyer, A.: Batch Verification of ECDSA Signatures. In *Progress in Cryptology - AFRICACRYPT 2012*, *Lecture Notes in Computer Science*, vol. 7374, ed. by A. Mitrokotsa; S. Vaudenay, Springer Berlin Heidelberg, 2012, ISBN 978-3-642-31409-4, pp. 1–18.

[94] Kelly, D.; Raines, R.; Baldwin, R.; Grimaila, M.; Mullins, B.: Exploring extant and emerging issues in anonymous networks: A taxonomy and survey of protocols and metrics. *Communications Surveys & Tutorials, IEEE*, vol. 14, no. 2, 2012: pp. 579–606.

[95] Kido, H.; Yanagisawa, Y.; Satoh, T.: An anonymous communication technique using dummies for location-based services. In *Pervasive Services, 2005. ICPS'05. Proceedings. International Conference on*, IEEE, 2005, pp. 88–97.

[96] Kitae, K.; Ikkwon, Y.; Seongan, L.; Daehun, N.: Batch Verification and Finding Invalid Signatures in a Group Signature Scheme. *International Journal of Network Security*, vol. 12, 2011: pp. 229–238.

[97] Kravitz, D. W.: Digital signature algorithm. Jul. 27 1993, uS Patent 5,231,668.

[98] Krumm, J.: A survey of computational location privacy. *Personal Ubiquitous Comput.*, vol. 13, no. 6, Aug. 2009: pp. 391–399, ISSN 1617-4909.

[99] Langheinrich, M.: A survey of RFID privacy approaches. *Personal and Ubiquitous Computing*, vol. 13, no. 6, Aug. 2009: pp. 413–421, ISSN 1617-4909.

[100] LEE, C.-C.; CHANG, T.-Y.; HWANG, M.-S.: A New Group Signature Scheme Based on the Discrete Logarithm. *Journal of Information Assurance and Security*, 2010.

[101] Li, J.; Rajan, A.: An Anonymous Attestation Scheme with Optional Traceability. In *Trust and Trustworthy Computing*, *Lecture Notes in Computer Science*, vol. 6101, 2010, pp. 196–210.

[102] Liang, X.; Cao, Z.; Shao, J.; Lin, H.: Short group signature without random oracles. In *Information and Communications Security*, Springer, 2007, pp. 69–82.

[103] Liang, X.; Lu, R.; Chen, L.; Lin, X.; Shen, X. S.: PEC: A privacy-preserving emergency call scheme for mobile healthcare social networks. *Journal of Communications and Networks*, vol. 13, no. 2, 2011: pp. 102–112.

[104] Libert, B.; Peters, T.; Yung, M.: Scalable group signatures with revocation. In *Advances in Cryptology–EUROCRYPT 2012*, Springer, 2012, pp. 609–627.

[105] Libert, B.; Vergnaud, D.: Group signatures with verifier-local revocation and backward unlinkability in the standard model. In *Cryptology and Network Security*, Springer, 2009, pp. 498–517.

[106] Lin, C.-H.; Hsu, R.-H.; Harn, L.: Improved DSA variant for batch verification. *Applied Mathematics and Computation*, vol. 169, no. 1, 2005: pp. 75 – 81, ISSN 0096-3003.

[107] Lin, H.-Y.; Tzeng, W.-G.: An efficient solution to the millionaire problem based on homomorphic encryption. In *Applied Cryptography and Network Security*, Springer, 2005, pp. 97–134.

[108] Lin, X.; Sun, X.; han Ho, P.; Shen, X.: GSIS: A Secure and Privacy Preserving Protocol for Vehicular Communications. In *IEEE Transactions on Vehicular Technology*, vol. 56, 2007, pp. 3442–3456.

[109] Lindell, Y.: Anonymous authentication. *Journal of Privacy and Confidentiality*, vol. 2, no. 2, 2007: p. 4.

[110] Lu, S.; Ostrovsky, R.; Sahai, A.; Shacham, H.; Waters, B.: Sequential Aggregate Signatures and Multisignatures Without Random Oracles. In *Advances in Cryptology - EUROCRYPT 2006*, *Lecture Notes in Computer Science*, vol. 4004, ed. by S. Vaudenay, Springer Berlin / Heidelberg, 2006, ISBN 978-3-540-34546-6, pp. 465–485.

[111] Luo, S.; Hu, J.; Chen, Z.: An Identity-Based One-Time Password Scheme with Anonymous Authentication. In *Networks Security, Wireless Communications and Trusted Computing, 2009. NSWCTC '09. International Conference on*, vol. 2, 2009, pp. 864 –867.

[112] Lysyanskaya, A.; Micali, S.; Reyzin, L.; Shacham, H.: Sequential aggregate signatures from trapdoor permutations. In *Advances in Cryptology â€“ EUROCRYPT 2004*, Springer-Verlag, 2004, pp. 74–90.

[113] Mahmoud, M.; Taha, S.; Misic, J.; Shen, X.: Lightweight Privacy-Preserving and Secure Communication Protocol for Hybrid Ad Hoc Wireless Networks. 2013.

[114] Malina, L.; Castellà-Roca, J.; Vives-Guasch, A.; Hajny, J.: Short-Term Linkable Group Signatures with Categorized Batch Verification. In *Foundations and Practice of Security*, Springer, 2013, pp. 244–260.

[115] Malina, L.; Clupek, V.; Martinasek, Z.; Hajny, J.; Oguchi, K.; Zeman, V.: Evaluation of Software-Oriented Block Ciphers on Smartphones. In *Foundations and Practice of Security*, Lecture Notes in Computer Science, Springer International Publishing, 2014, ISBN 978-3-319-05301-1, pp. 353–368, doi:10.1007/978-3-319-05302-8_22.

[116] Malina, L.; Hajny, J.: Accelerated Modular Arithmetic for Low- Performance Devices. In *34th International Conference on Telecommunications and Signal Processing (TSP 2011)*, 2011, ISBN 978-1-4577-1409- 2, pp. 1–5.

[117] Malina, L.; Hajny, J.: Privacy-preserving framework for geosocial applications. *Security and Communication Networks*, 2013.

[118] Malina, L.; Hajny, J.: Efficient Modular Multiplication for Programmable Smart-Cards. *Telecommunication Systems*, vol. 55, 2014: pp. 1–9, ISSN 1018-4864.

[119] Malina, L.; Hajny, J.; Martinasek, Z.: Efficient Group Signatures with Verifier-local Revocation Employing a Natural Expiration. In *SECRYPT*, 2013, pp. 555–560.

[120] Malina, L.; Hajny, J.; Zeman, V.: Group signatures for secure and privacy preserving vehicular ad hoc networks. In *Proceedings of the 8h ACM symposium on QoS and security for wireless and mobile networks*, ACM, 2012, pp. 71–74.

[121] Malina, L.; Hajny, J.; Zeman, V.: Trade-off between signature aggregation and batch verification. In *Telecommunications and Signal Processing (TSP), 2013 36th International Conference on*, IEEE, 2013, pp. 57–61.

[122] Malina, L.; Hajny, J.; Zeman, V.: Usability of Pairing-Based Cryptography on Smartphones. In *37th International Conference on Telecommunications and Signal Processing (TSP 2014)*, 2014, ISBN 978-1-4577-1409- 2, pp. 1–5.

[123] Malina, L.; Vives-Guasch, A.; Castellà-Roca, J.; Viejo, A.; Hajny, J.: Group Signatures with Categorized Batch Verification. *Telecommunication Systems*, 2014: pp. 1–16, ISSN 1018-4864.

[124] Malkhi, D.; Nisan, N.; Pinkas, B.; Sella, Y.; et al.: Fairplay-Secure two-party computation system. In *USENIX Security Symposium*, 2004, pp. 287–302.

[125] Martucci, L.; Ries, S.; Mühlhäuser, M.: Sybil-FreePseudonyms, Privacy and Trust: Identity Management in the Internet of Services. *Journal of Information Processing*, 2011.

[126] Mascetti, S.; Freni, D.; Bettini, C.; Wang, X. S.; Jajodia, S.: Privacy in geo-social networks: proximity notification with untrusted service providers and curious buddies. *The VLDB Journal*, vol. 20, no. 4, Aug. 2011: pp. 541–566, ISSN 1066-8888.

[127] Miller, V. S.: The Weil Pairing, and Its Efficient Calculation. *J. Cryptol.*, vol. 17, no. 4, Sep. 2004: pp. 235–261, ISSN 0933-2790.

[128] Miyaji, A.; Nakabayashi, M.; TAKANO, S.: New explicit conditions of elliptic curve traces for FR-reduction. 2001.

[129] Mostowski, W.; Vullers, P.: Efficient U-Prove Implementation for Anonymous Credentials on Smart Cards. 2011.

[130] Nácher, M.; Calafate, C. T.; Cano, J.-C.; Manzoni, P.: An overview of anonymous communications in mobile ad hoc networks. *Wireless Communications and Mobile Computing*, vol. 12, no. 8, 2012: pp. 661–675.

[131] Nakanishi, T.; Funabiki, N.: A short verifier-local revocation group signature scheme with backward unlinkability. *IEICE transactions on fundamentals of electronics, communications and computer sciences*, vol. 90, no. 9, 2007: pp. 1793–1802.

[132] Neven, G.: Efficient Sequential Aggregate Signed Data. In *Advances in Cryptology â€“ EUROCRYPT 2008*, *Lecture Notes in Computer Science*, vol. 4965, ed. by N. Smart, Springer Berlin / Heidelberg, 2008, ISBN 978-3-540-78966-6, pp. 52–69.

[133] Nguyen, L.; Safavi-Naini, R.: Efficient and provably secure trapdoor-free group signature schemes from bilinear pairings. In *Advances in Cryptology-ASIACRYPT 2004*, Springer, 2004, pp. 372–386.

[134] Nguyen, L.; Safavi-Naini, R.: Dynamic k-Times Anonymous Authentication. In *Applied Cryptography and Network Security*, *Lecture Notes in Computer Science*, vol. 3531, 2005, pp. 318–333.

[135] Pandey, A.; Tripathi, R.: Article:A Survey on Wireless Sensor Networks Security. *International Journal of Computer Applications*, vol. 3, no. 2, June 2010: pp. 43–49, published By Foundation of Computer Science.

[136] Pastuszak, J.; MichaĹ,ek, D.; Pieprzyk, J.; Seberry, J.: Identification of Bad Signatures in Batches. In *Public Key Cryptography*, *Lecture Notes in Computer Science*, vol. 1751, ed. by H. Imai; Y. Zheng, Springer Berlin, 2000, ISBN 978-3-540-66967-8, pp. 28–45.

[137] Pathan, A.-S. K.: *Security of self-organizing networks: MANET, WSN, WMN, VANET*. CRC press, 2010.

[138] Pedersen, T. P.: Non-interactive and information-theoretic secure verifiable secret sharing. In *Advances in Cryptology—CRYPTO'91*, Springer, 1992, pp. 129–140.

[139] Pereira, G. C. C. F.; Simplício, M. A., Jr.; Naehrig, M.; Barreto, P. S. L. M.: A family of implementation-friendly BN elliptic curves. *J. Syst. Softw.*, vol. 84, no. 8, Aug. 2011: pp. 1319–1326, ISSN 0164-1212.

[140] Petit, J.; Mammeri, Z.: Authentication and consensus overhead in vehicular ad hoc networks. *Telecommunication Systems*: pp. 1–14, ISSN 1018-4864, 10.1007/s11235-011-9589-y.

[141] Prasad, N. R.; Alam, M.; Ruggieri, M.: Light-weight AAA infrastructure for mobility support across heterogeneous networks. *Wireless Personal Communications*, vol. 29, no. 3-4, 2004: pp. 205–219.

[142] Puttaswamy, K. P. N.; Zhao, B. Y.: Preserving privacy in location-based mobile social applications. In *Proceedings of the Eleventh Workshop on Mobile Computing Systems & Applications*, HotMobile '10, New York, NY, USA: ACM, 2010, ISBN 978-1-4503-0005-6, pp. 1–6, doi:10.1145/1734583.1734585.

[143] Qin, B.; Wu, Q.; Domingo-Ferrer, J.; Zhang, L.: Preserving security and privacy in large-scale VANETs. In *Proceedings of the 13th international conference on Information and communications security*, ICICS'11, Springer-Verlag, 2011, ISBN 978-3-642-25242-6, pp. 121–135.

[144] Raya, M.; Hubaux, J.-P.: Securing vehicular ad hoc networks. *J. Comput. Secur.*, vol. 15, January 2007: pp. 39–68, ISSN 0926-227X.

[145] Raya, M.; Papadimitratos, P.; Hubaux, J.-P.: SECURING VEHICULAR COMMUNICATIONS. *Wireless Communications, IEEE*, vol. 13, no. 5, october 2006: pp. 8 –15, ISSN 1536-1284, doi:10.1109/WC-M.2006.250352.

[146] Reiter, M. K.; Rubin, A. D.: Anonymous Web Transactions with Crowds. *Commun. ACM*, vol. 42, no. 2, Feb. 1999: pp. 32–48, ISSN 0001-0782, doi:10.1145/293411.293778.

[147] Rivest, R. L.; Shamir, A.; Adleman, L.: A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, vol. 21, no. 2, 1978: pp. 120–126.

[148] Ruiz Vicente, C.; Freni, D.; Bettini, C.; Jensen, C.: Location-Related Privacy in Geo-Social Networks. *Internet Computing, IEEE*, vol. 15, no. 3, may-june 2011: pp. 20 –27, ISSN 1089-7801.

[149] Schnorr, C.-P.: Efficient signature generation by smart cards. *Journal of cryptology*, vol. 4, no. 3, 1991: pp. 161–174.

[150] Schroder, D.: How to Aggregate the CL Signature Scheme. In *ESORICS 2011, LNCS 6879*, Springer Berlin / Heidelberg, 2011, pp. 298–314.

[151] Scott, M.: Efficient Implementation of Cryptographic pairings. 2007.

[152] Sen, J.: Security and privacy issues in wireless mesh networks: A survey. In *Wireless Networks and Security*, Springer, 2013, pp. 189–272.

[153] Seys, S.; Preneel, B.: ARM: Anonymous routing protocol for mobile ad hoc networks. *International Journal of Wireless and Mobile Computing*, vol. 3, no. 3, 2009: pp. 145–155.

[154] Sgora, A.; Vergados, D. D.; Chatzimisios, P.: A survey on security and privacy issues in Wireless Mesh Networks. *Security and Communication Networks*, 2013.

[155] Shahandashti, S. F.; Safavi-Naini, R.: Threshold Attribute-Based Signatures and Their Application to Anonymous Credential Systems. In *Proceedings of the 2nd International Conference on Cryptology in Africa: Progress in Cryptology*, AFRICACRYPT '09, Berlin, Heidelberg: Springer-Verlag, 2009, ISBN 978-3-642-02383-5, pp. 198–216.

[156] Sterckx, M.; Gierlichs, B.; Preneel, B.; Verbauwhede, I.: Efficient Implementation of Anonymous Credentials on Java Card Smart Cards. In *1st IEEE International Workshop on Information Forensics and Security (WIFS 2009)*, London,UK: IEEE, 2009, pp. 106–110.

[157] Studer, A.; Shi, E.; Bai, F.; Perrig, A.: TACKing Together Efficient Authentication, Revocation, and Privacy in VANETs. In *SECON*, IEEE, 2009, ISBN 978-1-4244-2907-3, pp. 1–9.

[158] Sun, J.; Zhang, C.; Fang, Y.: A security architecture achieving anonymity and traceability in wireless mesh networks. In *INFOCOM 2008. The 27th Conference on Computer Communications. IEEE*, IEEE, 2008.

[159] Sweeney, L.: k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 10, no. 05, 2002: pp. 557–570.

[160] Taylor, D.; Davis, D.; Jillapalli, R.: Privacy concern and online personalization: The moderating effects of information control and compensation. *Electronic Commerce Research*, vol. 9, 2009: pp. 203–223, ISSN 1389-5753, doi:10.1007/s10660-009-9036-2.

[161] Teranishi, I.; Furukawa, J.; Sako, K.: k-Times Anonymous Authentication (Extended Abstract). In *Advances in Cryptology - ASIACRYPT 2004*, *Lecture Notes in Computer Science*, vol. 3329, ed. by P. J. Lee, 2004, pp. 81–95.

[162] van Thanh, D.; Jorstad, I.; Jonvik, T.; van Thuan, D.: Strong authentication with mobile phone as security token. In *Mobile Adhoc and Sensor Systems, 2009. MASS '09. IEEE 6th International Conference on*, oct. 2009, pp. 777 –782.

[163] Tsai, H.-W.: Aggregating data dissemination and discovery in vehicular Ad Hoc networks. *Telecommunication Systems*: pp. 1–11, ISSN 1018-4864, 10.1007/s11235-010-9404-1.

[164] Tsiounis, Y.; Yung, M.: On the security of ElGamal based encryption. In *Public Key Cryptography*, Springer, 1998, pp. 117–134.

[165] Tsudik, G.; Xu, S.: Accumulating composites and improved group signing. In *Advances in Cryptology-ASIACRYPT 2003*, Springer, 2003, pp. 269–286.

[166] Walters, J. P.; Liang, Z.; Shi, W.; Chaudhary, V.: Wireless sensor network security: A survey. In *Security in Distributed, Grid, and Pervasive Computing, Yang Xiao (Eds*, CRC Press, 2007, pp. 0–849.

[167] Wan, Z.; Ren, K.; Zhu, B.; Preneel, B.; Gu, M.: Anonymous user communication for privacy protection in wireless metropolitan mesh networks. *Vehicular Technology, IEEE Transactions on*, vol. 59, no. 2, 2010: pp. 519–532.

[168] Wasef, A.; Shen, X. S.: Efficient Group Signature Scheme Supporting Batch Verification for Securing Vehicular Networks. In *IEEE International Conference on Communications (ICC)*, 2010.

[169] Wei, L.; Liu, J.; Zhu, T.: On a Group Signature Scheme Supporting Batch Verification for Vehicular Networks. In *International Conference on Multimedia Information Networking and Security*, Los Alamitos, CA, USA: IEEE C. S., 2011, ISBN 978-0-7695-4559-2, pp. 436–440.

[170] Wen, F.; Susilo, W.; Yang, G.: A Secure and Effective Anonymous User Authentication Scheme for Roaming Service in Global Mobility Networks. *Wireless personal communications*, vol. 73, no. 3, 2013: pp. 993–1004.

[171] Werner, M.: Privacy-protected communication for location-based services. *Security and Communication Networks*, 2011: pp. n/a–n/a, ISSN 1939-0122, doi:10.1002/sec.330.

[172] Xie, B.; Kumar, A.; Zhao, D.; Reddy, R.: On secure communication in integrated heterogeneous wireless networks. *International Journal of Information Technology, Communications and Convergence*, vol. 1, no. 1, 2010: pp. 4–23.

[173] Yang, G.; Huang, Q.; Wong, D. S.; Deng, X.: Universal authentication protocols for anonymous wireless communications. *Wireless Communications, IEEE Transactions on*, vol. 9, no. 1, 2010: pp. 168–174.

[174] Yi, S.; Kravets, R.: Key management for heterogeneous ad hoc wireless networks. In *Network Protocols, 2002. Proceedings. 10th IEEE International Conference on*, IEEE, 2002, pp. 202–203.

[175] Yu, Y.; Zheng, X.; Sun, H.: An Identity Based Aggregate Signature from Pairings. *Journal of Networks*, vol. 6, no. 4, 2011.

[176] Zeadally, S.; Hunt, R.; Chen, Y.-S.; Irwin, A.; Hassan, A.: Vehicular ad hoc networks (VANETS): status, results, and challenges. *Telecommunication Systems*: pp. 1–25, ISSN 1018-4864, 10.1007/s11235-010-9400-5.

[177] Zhang, C.; Lu, R.; Lin, X.; Ho, P.-H.; Shen, X.: An Efficient Identity-Based Batch Verification Scheme for Vehicular Sensor Networks. In *INFOCOM*, IEEE, 2008, pp. 246–250.

[178] Zhang, L.; Ding, X.; Wan, Z.; Gu, M.; Li, X.-Y.: WiFace: a secure geosocial networking system using WiFi-based multi-hop MANET. In *Proceedings of the 1st ACM Workshop on Mobile Cloud Computing & Services: Social Networks and Beyond*, MCS '10, New York, NY, USA: ACM, 2010, ISBN 978-1-4503-0155-8, pp. 3:1–3:8.

[179] Zhang, L.; Wu, Q.; Solanas, A.; Domingo-Ferrer, J.: A Scalable Robust Authentication Protocol For Secure Vehicular Communications. In *IEEE Transactions on Vehicular Technology 59(4)*, 2010, pp. 1606–1617.

[180] Zhang, Y.; Liu, W.; Lou, W.: Anonymous communications in mobile ad hoc networks. In *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, vol. 3, IEEE, 2005, pp. 1940–1951.

[181] Zhou, S.; Lin, D.: Shorter verifier-local revocation group signatures from bilinear maps. In *Cryptology and Network Security*, Springer, 2006, pp. 126–143.

LIST OF ABBREVIATIONS

AAA	Authentication, Authorization and Accounting
AES	Advanced Encryption Standard
ATM	Asynchronous Transfer Mode
ART	Android RunTime
B	Data Bandwidth [bit/s]
BAN	Body Area Network
BBS04	the Boneh, Boyen and Shacham group signature scheme
BS04	the Boneh, Shacham group signature scheme
BDHP	Bilinear Diffie-Hellman Problem
BIDHP	Bilinear Inverse Diffie-Hellman Problem
BSDHP	Bilinear Square Diffie-Hellman Problem
BU	Backward Unlinkability
CDHP	Computational Diffie-Hellman Problem
CN	Cellular Network
CPU	Central Processing Unit
CZK	Computational Zero-Knowledge protocol
DDHP	Decision Diffie-Hellman Problem
DH	Diffie-Hellman Protocol
DL	Discrete Logarithm
DLP	Discrete Logarithm Problem
DLDHP	Decision Linear Diffie-Hellman Problem
DES	Data Encryption Standard
DoS	Denial of Service
DSA	Digital Signature Algorithm
ECC	Elliptic Curves Cryptography
ECDH	Elliptic Curve Diffie Hellman protocol
ECDSA	Elliptic Curve Digital Signature Algorithm

EDGE	Enhanced Data Rates for GSM Evolution
GLOMONET	Global Mobility Network
GM	Group Manager
GPRS	General Packet Radio Service
GSM	Global System for Mobile Communications
H	Hash function
HTTP	Hypertext Transfer Protocol
HVZK	Honest Verifier Zero Knowledge
IBGS	Identity Based Group Signature
ICT	Information and Communications Technology
ID	Identity
IP	Internet Protocol
IPS	Intrusion Prevention Systems
iOS	iPhone Operating System
IoT	Internet of Things
IrDA	Infrared Data Association
IZK	Interactive Zero-Knowledge protocol
LAN	Local Area Network
MAC	Message Authentication Code
MAN	Metropolitan Area Network
MANET	Mobile Ad hoc Network
MD5	Message-Digest algorithm
MIX	Mixes mechanism
MN	Mesh Network
MNT	Miyaji-Nakabayashi-Takano curves
MPLS	Multiprotocol Label Switching
MULTOS	Multi-application smart card Operating System
NFC	Near Field Communication

NIWI	Non-Interactive Witness-Indistinguishable
NIZK	Non-Interactive Zero-Knowledge protocol
OBUs	On Board Units
P	Prover
PAN	Personal Area Network
PBC	Pairing-Based Cryptography
PC	Personal Computer
PETs	Privacy Enhancing Technologies
PK	Proof of Knowledge
PKDL	Proof of Knowledge of Discrete Logarithm
PKI	Public Key Infrastructure
PZK	Perfect Zero-Knowledge protocol
QR	Quick Response codes
qSDHP	q-Strong Diffie-Hellman problem
RAM	Random-Access Memory
RFID	Radio-Frequency IDentification
RNG	Random Number Generator
RO	Random Oracle
RSA	Rivest, Shamir, Adleman
RSU	Road Side Units
SDH	Synchronous Digital Hierarchy
SDHP	Strong Diffie-Hellman Problem
SIM	Subscriber Identification Module
SHA	Secure Hash Algorithm
SONs	Self-Organizing Networks
SONET	Synchronous Optical Network
SP	Service Provider
SZK	Statistical Zero-Knowledge protocol

TA	Trusted Authority
TCP	Transmission Control Protocol
TDES	Triple Data Encryption Standard
TLS	Transport Layer Security Protocol
TTP	Third Trusted Party
U	User
UMTS	Universal Mobile Telecommunication System
V	Verifier / Vehicle(in protocol 2)
VANET	Vehicular Ad-Hoc Network
VLR	Verifier Local Revocation
VoIP	Voice-Over-IP
V2I	Vehicle to Infrastructure communication
V2V	Vehicle to Vehicle communication
WiMAX	Worldwide interoperability for Microwave Access
WMN	Wireless Mesh Network
WMMN	Wireless Metropolitan Mesh Network
WSN	Wireless Sensor Network
XDHP	eXternal Diffie-Hellman Problem
ZK	Zero-Knowledge
...	Other notation used in Protocol 2 and the framework are in Tables 8.1 and 9.1

A AUTHOR'S SELECTED PUBLICATION

Papers in impacted journals ISI JCR

1. Malina, L.; Hajny, J.: Efficient Modular Multiplication for Programmable Smart-Cards. *Telecommunication Systems*, 2014: p. 1-9, vol.55, no.4, ISSN 1018-4864.(IF: 1,027).

2. Malina, L. ; Hajny, J.: Privacy-preserving framework for geosocial applications. *Security and Communication Networks*, 2013. ISSN: 1939-0122. (IF: 0,311, online first).

3. Malina, L.; Vives-Guasch, A.; Castella-Roca, J.; Viejo A.; Hajny, J.: Efficient Group Signatures for Privacy-Preserving Vehicular Networks. *Telecommunication Systems*, 2014. (IF: 1,027, To Appear)

Papers in Lecture Notes in Computer Science

4. Malina, L.; Clupek, V.; Martinasek, Z.; Hajny, J.; Oguchi, K.; Zeman, V. Evaluation of Software- Oriented Block Ciphers on Smartphones. In Foundations and Practice of Security. Lecture Notes in Computer Science. Springer International Publishing, 2014. pp. 353-368. ISBN: 978-3-319-05301- 1. ISSN: 0302-9743.

5. Malina, L.; Castella-Roca, J.; Vives-Guasch, A.; Hajny, J. Short- Term Linkable Group Signatures with Categorized Batch Verification. In Foundations and Practice of Security. Lecture Notes in Computer Science. LNCS. Berlin: Springer- Verlag, 2013. pp. 244-260. ISBN: 978-3-642-37118- 9. ISSN: 0302-9743

6. Hajny, J.; Malina, L.; Martinasek, Z.; Tethal, O. Performance Evaluation of Primitives for Privacy-Enhancing Cryptography on Current Smart-cards and Smart-phones. In Data Privacy Management and Autonomous Spontaneous Security. Springer, 2014. pp. 17-33. ISBN: 978-3-642-54567- 2.

7. Hajny, J.; Malina, L. Unlinkable Attribute-Based Credentials with Practical Revocation on Smart-Cards. In Smart Card Research and Advanced Applications. Lecture Notes in Computer Science. LNCS. Berlin: Springer- Verlag, 2013. pp. 62-76. ISBN: 978-3-642-37287- 2. ISSN: 0302-9743

Conference Papers

8. Malina, L.; Hajny, J.; Zeman, V.: Usability of Pairing-Based Cryptography on Smartphones. In *37th International Conference on Telecommunications and Signal Processing (TSP 2014)*. 2014. p. 1-5. ISBN 978-1-4577-1409-2,

9. Malina, L.; Hajny, J.; Martinasek, Z. Efficient Group Signatures with Verifier- local Revocation Employing a Natural Expiration. In *Proceedings of the 10th International Conference on Security and Cryptography (SECRYPT 2013)*. 2013. pp. 555-560. ISBN: 978-989-8565-73-0.

10. Malina, L.; Hajny, J.; Zeman, V. Trade-off between Signature Aggregation and Batch Verification. In *36th International Conference on Telecommunications and Signal Processing (TSP 2013)*. 2013. pp. 57-62. ISBN: 978-1-4799-0403-7.

11. Malina, L.; Hajny, J. Efficient Security Solution for Privacy-Preserving Cloud Services. In *36th International Conference on Telecommunications and Signal Processing (TSP 2013)*. 2013.

pp. 23-28. ISBN: 978-1-4799-0403-7.

12. Hajny, J.; Malina, L.; Martinasek, Z.; Zeman, V. Privacy-preserving SVANETs: Privacy-preserving Simple Vehicular Ad- hoc Networks. In *Proceedings of the 10th International Conference on Security and Cryptography (SECRYPT 2013)*. 2013. pp. 267-274. ISBN: 978-989-8565-73-0.

13. Malina, L.; Hajny, J.; Zeman, V. Group signatures for secure and privacy preserving vehicular ad hoc networks. In *Proceedings of the 8h ACM symposium on QoS and security for wireless and mobile networks - Q2SWinet ' 12*. New York, NY, USA: ACM, 2012. pp. 71-74. ISBN: 978-1-4503-1619-4.

14. Hajny, J.; Malina, L. Anonymous Credentials with Practical Revocation. In *Proceedings of the First AESS European Conference on Satellite Telecommunications.* 2012. p. 1-6. ISBN: 978-1-4673-4688-7.

15. Cervenka, V.; Komosny, D.; Malina, L.; et al.: Energy Efficient Public Key Cryptography in Wireless Sensor Networks. In *Innovations and Advances in Computer, Information, Systems Sciences, and Engineering.* USA, Berkeley: Springer New York, 2012. p. 497-509. ISBN: 978-1-4614-3534-1.

16. Malina, L.; Hajny, J.: Accelerated Modular Arithmetic for Low-Performance Devices. In *34th International Conference on Telecommunications and Signal Processing (TSP 2011)*. 2011. p. 1-5. ISBN 978-1-4577-1409-2.

17. Malina, L.; Zukal, M.: Secure Authentication and Key Establishment in the SIP Architecture. In *34th International Conference on Telecommunications and Signal Processing (TSP)*. 2011. p. 1-5. ISBN 978-1-4577-1409-2.

18. Hajny, J.; Malina, L.; Zeman, V.: PRACTICAL ANONYMOUS AUTHENTICATION: Designing Anonymous Authentication for Everyday Use. In *Proceedings of the Secrypt 2011*, 2011,p. 405-408., ISBN 978-989-8425-18-8.

19. Malina, L.; Hajny, J.: Secure Authentication in Privacy Protection Systems. In *Proceedings of the 16th conference Student EEICT 2010*, 2010, ISBN 978-80-214-4079-1, p. 1-3.

20. Hajny, J.; Malina, L.; Pelka, T.: Zero-Knowledge for Anonymous Authentication. In *33th International Conference on Telecommunications and Signal Processing (TSP 2010)*. 2010. p. 1-6. ISBN 978-963-88981-0-4.

Papers in Non-impacted Journals

21. Malina, L.; Zeman, V.: Comprehensive Security in SIP. *Elektrorevue - Internetový Časopis (http://www.elektrorevue.cz)*, vol. 2, no. 1, 2011: p. 1-8, ISSN 1213-1539.

22. Malina, L.; Zukal, M.: Practical Security in the SIP Architecture. *Elektrorevue (http://www.elektrorevue.cz)*, vol. 2, no. 4, 2011: p. 1-7, ISSN 1213-1539.

23. Hajny, J.; Malina, L.: Implementation Results of Anonymous Authentication Scheme. *Elektrorevue (http://www.elektrorevue.cz)*, vol. 86, 2010: p. 1-8, ISSN 1213-1539.

Curriculum Vitae

Lukáš Malina

Affiliation:

Address:	Brno University of Technology, Brno, Czech Republic
E-mail:	malina@feec.vutbr.cz, lukas.malina85@gmail.com
GSM:	+420 605 339 791

Education:

2010-present	Brno University of Technology, Ph.D. Faculty of Electrical Engineering and Communication Specialization: Teleinformatics Theme of thesis: Privacy preserving cryptographic protocols for secure heterogeneous networks
2011-2012	Universitat Rovira i Virgili, Spain, Ph.D. exchange stay Department of Computer Engineering and Mathematics Specialization: Applied Cryptography Research in: Pairing-based group signatures
2008-2010	Brno University of Technology, MSc. (Master's degree with honour) Faculty of Electrical Engineering and Communication Specialization: Communications and Informatics Theme of thesis: Internet Privacy Protection
2005-2008	Brno University of Technology, BSc. (Bachelor's degree with honour) Faculty of Electrical Engineering and Communication Specialization: Teleinformatics Theme of thesis: Modern Computer's Viruses

Work experience:

2011-present	Junior researcher, Brno University of Technology
2010-present	Cryptography course lab instructor, Brno University of Technology

Project participation:

2013-2015	TA03010818: Application of modern cryptographical methods to increase communication security in telematics systems Principal investigator: doc. Ing. Václav Zeman, Ph.D.

2012-2014	TA02011260: Cryptographic system for the protection of electronic identity, Principal investigator: Prof. Ing. Kamil Vrba, CSc.
2012-2014	FEKT-S-11-15: Research in Electronic Communication Systems Principal investigator: Prof. Ing. Kamil Vrba, CSc.
2012-2014	FR-TI3/170: Integration server with cryptographic protection Principal investigator: Prof. Ing. Kamil Vrba, CSc.
2013	ED2.1.00/03.0072, Center of Sensor, Information and Communication Systems, Principal investigator: Prof. Dr. Ing. Zbyněk Raida
2010-2013	FR-TI2/679: Media-informatics system supporting advanced multimedia services, Principal investigator: doc. Ing. Václav Zeman, Ph.D,
2012	1333/2012/G1: Inclusion of the issue of privacy protection using programmable smart cards to cryptography courses Principal investigator: Ing. Lukáš Malina

Designated reviewer:

- Security and Communication Networks – Wiley, USA, ISSN: 1939-0122
- 35th International Conference on Telecommunications and Signal Processing (TSP 2012)
- 36th International Conference on Telecommunications and Signal Processing (TSP 2013)
- 37th International Conference on Telecommunications and Signal Processing (TSP 2014)
- Elektrorevue – Internet Electrotechnics Magazine, ISSN 1213-1539
- IJATES – Internet All-Electronic Journal, ISSN 1805-5443

Selected invited presentations:

- 2014, Workshop: Cyber Security and Today's Communication Technologies, Brno, CZ
- 2013, Tampere University of Technology, Tampere, Finland

Publication activities:

- Papers published in impact factor journals: **3**
- Papers published in other journals: **4**
- Papers published in international conferences: **18**
- Papers published in domestic conferences: **6**
- Papers indexed in WoS: **7**
- Papers indexed in Scopus: **17**
- H-index according to WoS: **2**
- H-index according to Scopus: **2**
- Number of released products: **7**

Last actualization: June 13, 2014